A COMPUTATIONALLY EFFICIENT STRATEGY FOR TIME-FRACTIONAL DIFFUSION-REACTION EQUATIONS

ROBERTO GARRAPPA AND MARINA POPOLIZIO

Abstract. An efficient strategy for the numerical solution of time-fractional diffusion-reaction problems is devised. A standard finite difference discretization of the space derivative is initially applied which results in a linear stiff term. Then a rectangular product-integration (PI) rule is implemented in an implicit-explicit (IMEX) framework in order to implicitly treat this linear stiff term and handle in an explicit way the non-linear, and usually non-stiff, term.

The kernel compression scheme (KCS) is successively adopted to reduce the overload of computation and storage need for the persistent memory term. To reduce the computational effort the semidiscretized problem is described in a matrix-form, so as to require the solution of Sylvester equations only with small matrices.

Theoretical results on the accuracy, together with strategies for the optimal selection of the main parameters of the whole strategy, are derived and verified by means of numerical experiments carried out in two-dimensional domains. The computational advantages with respect to other approaches are also shown and some applications to the detection of pattern formation are illustrated at the end of the paper.

1. Introduction

Diffusion-Reaction (DR) systems represent an important class of partial differential equations (PDEs) used to describe models in many branches of biology, chemistry, physics etc. The theoretical analysis of DR equations is still an active research subject whose development has gone hand in hand with the development of numerical methods for their resolution (we just refer to [1; 2; 3] and references therein). In recent times, DR systems have attracted much interest as a prototype model for pattern formation; their solutions can indeed present spatially inhomogeneous distributions showing "Turing patterns" like stripes, hexagons, spirals, targets, hexagons etc. [4; 5; 6].

Recent researches indicate that classical DR systems may be inadequate to model many real situations. To fill this gap, Fractional Diffusion-Reaction (FDR) equations have been proposed; they generalize DR systems by considering a time-derivative with non-integer (i.e. positive real) order. The presence of this real parameter represents an additional degree of freedom which allows to better model real life problems. Speaking in general, Fractional Differential Equations (FDEs), namely differential equations involving one or more derivatives of non-integer order, are capturing an increasing attention since their ability to incorporate memory effects usually observed in anomalous, heterogeneous or complex materials [7; 8; 9; 10; 11; 12; 13]. FDR equations are commonly used to model the phenomenon of anomalous diffusion [14; 15; 16], different self-organization phenomena and specific nonlinear effects [17] or continuous-time random walk model with temporal memory and sources [18].

Founded by the PRIN project MIUR 2017JYCLSF_003 and by the INdAM project GNCS 2020.

Accepted version of the paper R.Garrappa, M.Popolizio, *A computationally efficient strategy for time-fractional diffusion-reaction equations*, Comput. Math. with Appl. **(2022)**, 116, 181-193, doi:10.1016/j.camwa.2021.05.027

In this work we propose an efficient numerical strategy to solve a FDR equation

$$ {}^{\mathrm{C}}\mathcal{D}_t^{\alpha} u = d\Delta u + f(u) \tag{1.1}$$

where Δu is the Laplacian operator, d a diffusion parameter and $f(u)$ a non-linear non-stiff reaction term. With ${}^{\mathrm{C}}\mathcal{D}_t^{\alpha}$ we denote the time fractional derivative of order $0 < \alpha < 1$ regularized in the Dzhrbashyan-Caputo way (usually simply known as Caputo's fractional derivative) [10; 11; 19; 20; 21] defined, for a sufficiently regular function $y(t)$, as

$$ {}^{\mathrm{C}}\mathcal{D}_t^{\alpha} y(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} y'(\tau) \mathrm{d}\tau, \quad t > 0, $$

where $\Gamma(z) = \int_0^{\infty} t^{z-1} \mathrm{e}^{-t} \mathrm{d}t$ is the Euler-Gamma function.

For simplicity we restrict our attention to two-dimensional domains and therefore $u = u(x, y, t)$ for $(x, y) \in \Omega \subset \mathbb{R}^2$ and $t \geq 0$. To ensure the uniqueness of the solution, equation (1.1) must be coupled with the initial condition

$$ u(x, y, 0) = u_0(x, y), \quad (x, y) \in \Omega, $$

for some assigned function u_0, and with boundary conditions of Dirichlet or Neumann type.

We focus here on subdiffusive processes of order $0 < \alpha < 1$; however, the extension to problems with fractional higher-order derivatives, in particular superdiffusive problems with $1 < \alpha < 2$, involves just minor issues (related to the presence of additional initial conditions) but it does not affect computational issues which are the main target of this work.

One of the main difficulties in solving FDEs is the presence of a persistent memory which is typical of fractional, and more generally non-local [22; 23; 24; 25], operators. This means that computing the solution of (1.1) at a given time t requires the knowledge and use of its history on the whole interval $[0, t]$, resulting in an extremely demanding need of storage memory and computational resources. For these reasons efficient numerical strategies are mandatory to perform simulations in an acceptable time and with a reasonable memory occupation. The first aim of the proposed strategy is therefore to keep storage and computation low even with the challenging problem of FDR systems.

To numerically solve (1.1) we consider here the Method of Lines (MOL) by which the original time-dependent PDE is recast as a system of ordinary differential equations (ODEs) (the *semi-discrete* system) after the discretization, by finite differences, of the Laplacian operator. Since the large eigenvalues of the discretization matrix, semi-discrete diffusion problems are usually *stiff*, thus to force the use of implicit methods for their time integration [1]. Moreover, the discretization of a 2-dimensional operator may lead serious problems related to the large size; indeed, if we fix a spatial grid with N_x nodes in the x-direction and N_y in the y-direction, then the size of the *semi-discrete* system is $\mathcal{O}\big(N_x \cdot N_y\big)$.

Recently Simoncini and coauthors [3; 26; 4; 27; 28] proposed matrix-equation-based strategies to reduce the computational effort involved by these large systems. As a particularly challenging example, we cite the numerical approximation of Turing patterns of FDR systems. This issue is extremely demanding since identifying the spatial structure of these patterns may require mesh-grids of very large size.

Regarding the time integration of the resulting system of FDEs, a longtime integration may be necessary (for instance to capture the asymptotic pattern formation in Turing patterns' detection). Additionally, a small time step-size h may be needed to capture an initial transient regime. These aspects are particularly challenging for

FDEs since, because of the persistent memory, it is necessary to save all the previously computed approximations. If N_t denotes the number of time grid points, the whole amount of memory storage required for the history of the solution is $\mathcal{O}\big(N_t \cdot N_x \cdot N_y\big)$. Not to mention that the number of floating-point operations with a PI rule implemented in a standard way would be proportional to $\mathcal{O}\big((N_t)^2\big)$ (or to $\mathcal{O}\big(N_t\big(\log_2 N_t\big)^2\big)$ when efficient methods [29; 30] are used), which may be not affordable.

It is therefore necessary to apply a suitable strategy which reduces both storage and computational needs. In this paper we propose a properly combined strategy mixing different approaches:

- a matrix approach to describe the semi-discretized problem: this expedient considerably reduces the computational effort since the most expensive part of the procedure is reduced to the solution of Sylvester equations with small matrices;
- product-integration (PI) rules for the time integration of FDEs in an IMEX framework to limit the implicit method just to the stiff part;
- Kernel Compression Scheme (KCS) to treat the persistent-memory typical of FDEs.

The paper is organized as follows: in Section 2 we address the semi-discretization of (1.1) by analyzing both a vector and a matrix formulation. In both cases the original FDR equation is transformed into a system of FDEs. For its numerical solution, in Section 3 we describe an IMEX version of a rectangular PI rule: in practice, a rectangular PI rule is proposed which treats the non linear part with explicit solvers and the stiff part with implicit methods. The numerical issues are addressed, while in Section 4 the accuracy and convergence properties are studied. In Section 5 we describe the KCS to effectively treat persistent-memory problems. Following, Section 6 presents numerical tests to show the accuracy and the efficiency of the proposed scheme; some numerical tests concerning the application of this strategy in pattern's detection are also presented. Some concluding remarks are provided at the end of the paper.

2. Semi-discretization in space: vector and matrix approaches

To apply the MOL method, in the FDR equation (1.1) just spatial derivatives are discretized and time variable are left continuous. To this purpose we fix a grid on the space domain by selecting N_x nodes $x_1, \dots, x_{N_x}$ in the x-direction and N_y nodes $y_1, \dots, y_{N_y}$ in the y-direction, with h_x and h_y the corresponding mesh sizes.

Moreover, we also fix a grid $t_n = nh$, $n = 0, \dots, N_t$, with step-size $h > 0$ and $T = N_t h$ on the time domain $[0, T]$ to proceed later with the time integration.

2.1. Vector approach.

To represent the exact solution u we can use the vector $\mathbf{u}(t) \in \mathbb{R}^{N_x N_y}$ whose entries are an approximation to u at the nodes, having used a lexicographic order for the nodes (x_i, y_j). Then we denote with

$$(\mathbf{u}_n)_{ij} \approx u(x_i, y_j, t_n).$$

To approximate the Laplacian we opt for the finite difference scheme: in particular, we use the second order difference formula [1]

$$\frac{1}{h_x^2}\Big(u(x - h_x, y, t) - 2u(x, y, t) + u(x + h_x, y, t)\Big) = \partial_{xx} u(x, y, t) + \mathcal{O}\big(h_x^2\big).$$

At this purpose we introduce the matrix $T_x = h_x^{-2}\text{diag}(1,-2,1) \in \mathbb{R}^{N_x \times N_x}$ which approximates the second order derivative in the x-direction. More precisely, to the matrix T_x we need to add a matrix term BC which takes into account the given boundary conditions. Analogously, the matrix $T_y \in \mathbb{R}^{N_y \times N_y}$, with h_y as scaling factor, approximates the second order derivative in the y-direction and incorporates the boundary conditions.

Then $\Delta \mathbf{u} \approx A\mathbf{u}$ where

$$A = I_y \otimes T_x + T_y \otimes I_x \in \mathbb{R}^{N_x N_y \times N_x N_y},$$

the symbol $\otimes$ denotes the Kronecker product, $I_x \in \mathbb{R}^{N_x \times N_x}$ and $I_y \in \mathbb{R}^{N_y \times N_y}$ are the identity matrices.

The numerical issue to face here is that the matrix A has large eigenvalues; in particular, the spectral radius of A is $\mathcal{O}(\tilde{h}^{-2})$ where $\tilde{h} = \min\{h_x, h_y\}$. For this reason, semi-discrete diffusion problems are usually classified as *stiff* problems, thus to require the use of implicit methods for their time integration [1].

After this semidiscretization, the equation (1.1) is reformulated as a system of FDEs

$$\begin{cases} {}^{\text{C}}\mathcal{D}_t^\alpha \mathbf{u}(t) = A\mathbf{u}(t) + f(\mathbf{u}(t)) \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases} \tag{2.1}$$

2.2. **Matrix approach.** Matrix-oriented strategies have been recently proposed in [3; 4] to drastically reduce the computational effort when solving PDEs. We describe here this approach which turns out to be very effective also in our context.

To represent the exact solution u we resort to a matrix $U(t) \in \mathbb{R}^{N_x \times N_y}$ such that

$$U_{i,j}(t) \approx u(x_i, y_j, t).$$

In this way

$$\Delta U \approx T_x U + U T_y$$

with T_x and T_y as before.

With this matrix notation, the equation (1.1) is reformulated as a system of FDEs

$$\begin{cases} {}^{\text{C}}\mathcal{D}_t^\alpha U(t) = T_x U(t) + U(t) T_y + F(U(t)) \\ U(0) = U_0 \end{cases} \tag{2.2}$$

where $(F(U(t)))_{i,j} = f(U_{i,j}(t))$ for $i = 1, \ldots, N_x$ and $j = 1, \ldots, N_y$.

After the discretization of the FDE (1.1) along the space variables we can rewrite it as (2.1) if the vector approach is used or in the form (2.2) if the matrix approach is considered. However, for convenience they both can be written in the general form

$$\begin{cases} {}^{\text{C}}\mathcal{D}_t^\alpha U(t) = G(t, U(t)) \\ U(0) = U_0 \end{cases}, \quad G(t, U(t)) = L(U(t)) + F(U(t)) \tag{2.3}$$

where $L(U(t))$ differs according to the vector or matrix approach employed to collect the semi-discretized variables $u(x_i, y_j, t)$, namely

$$L(U(t)) = \begin{cases} AU(t) & \text{with } U(t) \in \mathbb{R}^{N_x N_y} \\ T_x U(t) + U(t) T_y & \text{with } U(t) \in \mathbb{R}^{N_x \times N_y} \end{cases}$$

with $F(U(t))$ defined accordingly.

3. A product-integration rule in IMEX framework

Product-Integration (PI) rules are some of the most employed discretization techniques for time-fractional differential equations. They were introduced in 1954 by Young [31] to solve second-kind weakly-singular Volterra integral equations and successively applied to FDEs as well [32; 33]; the solution of (2.3) can be indeed formulated as [19; 10]

$$U(t) = U_0 + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} G(\tau, U(\tau)) \mathrm{d}\tau. \tag{3.1}$$

Let us now consider the numerical approximation of (3.1) by means of the PI rule (we refer to [34] for a thorough description). By referring to the time grid defined in Section 2, to approximate the solution at a value t_n we consider the equivalent expression

$$U(t_n) = U_0 + \frac{1}{\Gamma(\alpha)} \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} (t_n-\tau)^{\alpha-1} G(\tau, U(\tau)) \mathrm{d}\tau;$$

then, in each of the sub-intervals $[t_k, t_{k+1}]$, interpolating polynomials are used to approximate the vector field $G(\tau, U(\tau))$. Rectangular PI rules are the simplest scheme in this framework and they are obtained after the replacement of the vector field with zero-degree polynomials (i.e., constant values). Thus, whenever the value $G(t_k, U_k)$ is used to approximate $G(\tau, U(\tau))$, with $U_k \approx U(t_k)$, the *explicit rectangular PI rule* is obtained

$$U_n = U_0 + h^\alpha \sum_{k=0}^{n-1} b^{(\alpha)}_{n-k-1} G(t_k, U_k),$$

whilst when the approximation $G(\tau, U(\tau)) \approx G(t_{k+1}, U_{k+1})$ is alternatively used, the *implicit rectangular PI rule* is instead derived

$$U_n = U_0 + h^\alpha \sum_{k=1}^{n} b^{(\alpha)}_{n-k} G(t_k, U_k).$$

In both cases the weights b_n are easily evaluated after a standard exact integration and they are given by

$$b^{(\alpha)}_n = \frac{1}{\Gamma(\alpha)} \int_{-1}^{0} (n-\tau)^{\alpha-1} \mathrm{d}\tau = \frac{(n+1)^\alpha - n^\alpha}{\Gamma(\alpha+1)}. \tag{3.2}$$

Remark 3.1. Using higher-degree polynomials in PI rules usually does not lead to recognizable improvements; the lack of smoothness at the origin of the exact solution of FDEs (see [35; 36; 37; 38]) does not in general allow the achievement of high order of convergence. This is well-known since Dixon found that a convergence order less than 2 must be expected under reasonable smoothness properties [39] (see also [33]). In addition, the larger computational difficulties and the deterioration of stability properties [40] in consequence of the increase of the polynomial degree, make rectangular PI rules a satisfactory compromise.

In the presence of non-linearity, large systems of FDEs may demand an exceedingly high computational cost when an implicit scheme is used; conversely, explicit schemes do not appear suitable since the stiffness resulting from the discretization of the diffusive term. An Implicit-Explicit (IMEX) strategy is surely more advantageous for problems in which linear stiff and non-linear non-stiff terms coexist. The main idea of

IMEX is to apply the implicit scheme just to the linear stiff term and treat explicitly the non-linear non-stiff term.

The IMEX approach we propose for the time integration of the semidiscretized problem (2.3) consists in considering the implicit rectangular PI rule for the stiff linear term and its explicit version for the non-linear term. The following *rectangular PI IMEX* scheme results

$$U_n = U_0 + h^\alpha \sum_{k=1}^{n} b_{n-k}^{(\alpha)} L(U_k) + h^\alpha \sum_{k=0}^{n-1} b_{n-k-1}^{(\alpha)} F(U_k), \tag{3.3}$$

which, since $b_0^{(\alpha)} = 1/\Gamma(\alpha+1)$, can be rewritten as

$$U_n - \frac{h^\alpha}{\Gamma(\alpha+1)} L(U_n) = U_0 + h^\alpha b_{n-1}^{(\alpha)} F(U_0) + h^\alpha \sum_{k=1}^{n-1} \Big[b_{n-k}^{(\alpha)} L(U_k) + b_{n-k-1}^{(\alpha)} F(U_k) \Big]. \tag{3.4}$$

3.1. **Numerical aspects.** The scheme (3.4) is clearly implicit; however deep differences arise between the vector and the matrix form. Indeed, if the former is used, the numerical evaluation of (3.4) requires just the solution of the linear systems

$$\mathcal{A} U_n = \mathcal{B}_n, \quad \mathcal{A} = I - \frac{h^\alpha}{\Gamma(\alpha+1)} A \tag{3.5}$$

with the known term $\mathcal{B}_n$ collecting all terms depending on the known values $U_0, \ldots, U_{n-1}$, that is, the memory of the equation.

For the matrix form (2.2) the left hand side of (3.4) is instead $U_n - T_x U_n - U_n T_y$ and the numerical computation of (3.4) involves the solution of a Sylvester matrix equation which can be written in compact form as

$$\mathcal{T}_x U_n + U_n \mathcal{T}_y = \mathcal{B}_n, \quad \mathcal{T}_x = I - \frac{h^\alpha}{\Gamma(\alpha+1)} T_x, \quad \mathcal{T}_y = -\frac{h^\alpha}{\Gamma(\alpha+1)} T_y \tag{3.6}$$

and $\mathcal{B}_n$ the matrix version of the previous known term.

The solution to (3.6) exists and is unique if $\mathcal{T}_x$ and $-\mathcal{T}_y$ have disjoint spectra [41] and this condition is trivially verified in our case.

3.2. **Implementation issues.** At each time step the PI IMEX method (3.4) requires the solution of the *Sylvester* equation (3.6) of dimension $N_x \times N_y$ when the space semidiscretization is written in matrix form or the solution of the linear system (3.5) of dimension $N_x N_y \times N_x N_y$ involved in the vector approach. To have an idea of these dimensions, we assume to fix 400 points in the x and y directions; this choice leads to a Sylvester equation of dimension 400×400 or to a linear system of dimension 160000×160000 if the vector form is used. However, as stressed in [4], the implementation of both approaches can widely benefit from the fact that the coefficient matrices do not change as the iteration proceeds.

We give here some details to show how the implementation complexity simplifies, by following the implementation proposed in [4]. To solve the Sylvester equation (3.6) the "small" matrices $\mathcal{T}_x$ and $\mathcal{T}_y$ can be factorized just once [3; 41; 42]. In particular, we assume that $\mathcal{T}_x$ and $\mathcal{T}_y$ are diagonalizable. In fact, the matrices T_x and T_y are symmetric, except for the low-rank BC terms. This implies that they are likely to be diagonalizable (at worst considering a very small perturbation, that was never needed in our tests) because the diagonalizable matrices form a dense set. This property is

then immediately extended to $\mathcal{T}_x$ and $\mathcal{T}_y$. So we determine two nonsingular matrices S_x and S_y and two diagonal matrices Λ_x and Λ_y such that

$$\mathcal{T}_x = S_x \Lambda_x S_x^{-1}, \quad \mathcal{T}_y = S_y \Lambda_y S_y^{-1}.$$

Then, for each n we compute

$$Y = L \circ (S_x^{-1} \mathcal{B}_n S_y)$$

with the matrix $L \in \mathbb{R}^{N_x \times N_y}$ such that

$$L_{ij} = \left(\left(1 - \frac{h^\alpha}{\Gamma(\alpha+1)} (\Lambda_x)_{ii} \right) + \left(- \frac{h^\alpha}{\Gamma(\alpha+1)} (\Lambda_y)_{jj} \right) \right)^{-1}$$

and $\circ$ denoting the element-by-element (or Hadamard) product. Then the solution is $U_n = S_x Y S_y^{-1}$.

Regarding the solution of the linear system (3.5), we recall that the coefficient matrix $\mathcal{A}$ has dimension $N_x N_y \times N_x N_y$; however, its very sparse and structured nature can be profitably exploited to compute a factorization of $\mathcal{A}$: this is very effective since the factorization is done just once while the factors are used to solve the N_t systems required for the time integration in $[0, T]$. In particular, for our numerical tests we performed an LU factorization with pivoting.

4. Accuracy of the PI IMEX method

To study the accuracy of the rectangular PI IMEX scheme (3.4) we have first to mention that the exact solution of the nonlinear FDE (2.3) expands in mixed (i.e. integer and fractional) powers [36]

$$U(t) = U_0 + \sum_{k,\ell \in \mathbb{N}} t^{k+\ell\alpha} U_{k,\ell} = U_0 + t^\alpha U_{0,1} + \mathcal{O}(t) \tag{4.1}$$

for some $U_{k,\ell}$ independent of t.

We now show that the PI IMEX rule has first order of convergence, by closely following the approach proposed in [39] (and in line with similar results in [33]) for trapezoidal PI rules.

Before stating our main result we need some preliminary matters. We first recall the following discrete Gronwall inequality.

Lemma 4.1 ([39])**.** *Let $0 < \alpha < 1$ and $h > 0$; for $n = 0, \dots, N$ let $t_n = nh$ and v_n be a sequence of non-negative real numbers such that*

$$v_n \le V_1 + V_2 t_n^{\alpha-1} + V_3 h^\alpha \sum_{k=0}^{n-1} v_k (n-k)^{\alpha-1},$$

for some non-negative constants V_1, V_2 and a positive constant V_3 independent of h. Then for any $\tau > 0$ there exists a constant $C = C(\tau)$ such that

$$v_n \le C \left(V_1 + V_2 t_n^{\alpha-1} \right)$$

for any $n = 0, 1, \dots, N$ and $Nh \le \tau$.

We also need the following result about the rectangular PI weights.

Lemma 4.2. *Let $n \ge 0$ and the sequence $b_n^{(\alpha)}$ be defined according to (3.2). Then*

$$b_n^{(\alpha)} + b_{n-1}^{(\alpha)} \le \frac{2^{2-\alpha}}{\Gamma(\alpha)} n^{\alpha-1}.$$

Proof. From (3.2) it is easy to see that

$$b_n^{(\alpha)} + b_{n-1}^{(\alpha)} = \frac{1}{\Gamma(\alpha+1)} \int_{-1}^{1} (n-\tau)^{\alpha-1} \mathrm{d}\tau = \frac{2}{\Gamma(\alpha)} (n-\xi)^{\alpha-1}, \quad \xi \in [-1,1],$$

where the last equality is obtained by a straightforward application of the mean value theorem for integrals. The proof now immediately follows since $(n-\xi)^{\alpha-1} \le 2^{1-\alpha} n^{\alpha-1}$ whenever $\xi \in [-1,1]$. □

The convergence properties can be now stated by means of the following result where $\|\cdot\|$ denotes any vector-matrix norm.

Proposition 4.3. *Let $U(t_n)$ denote the exact solution of (2.3) at $t_n = nh$ and U_n its approximation computed by the rectangular PI IMEX scheme (3.3). If F is Lipschitz continuous then there exists a constant C independent of h such that for sufficiently small h*

$$\|U(t_n) - U_n\| \le C(h^{\alpha+1} t_n^{\alpha-1} + h t_n^{\alpha}), \quad n = 0, \dots, N.$$

Proof. We start by writing the exact solution of (2.3) as

$$U(t_n) = U_0 + h^\alpha \sum_{k=1}^{n} b_{n-k}^{(\alpha)} L(U(t_k)) + h^\alpha \sum_{k=0}^{n-1} b_{n-k-1}^{(\alpha)} F(U(t_k)) + T_n,$$

where $T_n = I_{n,0} + I_{n,1} + \cdots + I_{n,n-1}$ is the quadrature error with

$$I_{n,k} = \frac{1}{\Gamma(\alpha)} \int_{t_k}^{t_{k+1}} (t_n - \tau)^{\alpha-1} \Big[L(U(\tau)) - L(U(t_{k+1})) + F(U(\tau)) - F(U(t_k)) \Big] \mathrm{d}\tau.$$

To bound the error $E_n = \|U(t_n) - U_n\|$, we subtract (3.3) from the above representation of the exact solution; then, if we denote with K the Lipschitz constant of F, it is

$$E_n \le h^\alpha H \sum_{k=1}^{n} b_{n-k}^{(\alpha)} E_k + h^\alpha K \sum_{k=0}^{n-1} b_{n-k-1}^{(\alpha)} E_k + \|T_n\|$$

with H denoting $\|A\|$ in the vector case or $\|T_x\| + \|T_y\|$ in the matrix one.

Then, since $E_0 = 0$, for sufficiently small h we may assume $\frac{h^\alpha}{\Gamma(\alpha+1)} H < 1$ and equivalently write

$$E_n \le h^\alpha \sum_{k=1}^{n-1} c_{n-k}^{(\alpha)} E_k + \hat{T}_n$$

with

$$c_{n-k}^{(\alpha)} = \frac{H b_{n-k}^{(\alpha)} + K b_{n-k-1}^{(\alpha)}}{1 - \frac{h^\alpha}{\Gamma(\alpha+1)} H} \le \frac{\max\{H,K\}}{1 - \frac{h^\alpha}{\Gamma(\alpha+1)} H} \left(b_{n-k}^{(\alpha)} + b_{n-k-1}^{(\alpha)} \right), \quad \hat{T}_n = \frac{\|T_n\|}{1 - \frac{h^\alpha}{\Gamma(\alpha+1)} H}.$$

Using Lemma 4.2 we get

$$c_{n-k}^{(\alpha)} \le \frac{\max\{H,K\}}{1 - \frac{h^\alpha}{\Gamma(\alpha+1)} H} \frac{2^{2-\alpha}}{\Gamma(\alpha)} (n-k)^{\alpha-1} = C_\alpha (n-k)^{\alpha-1}$$

and hence

$$E_n \le h^\alpha C_\alpha \sum_{k=1}^{n-1} (n-k)^{\alpha-1} E_k + \hat{T}_n. \tag{4.2}$$

To study the term $\hat{T_n}$, i.e. the scaled truncation error T_n, we analyze each term $I_{n,k}$ by noting that

$$\|I_{n,k}\| \leq \frac{1}{\Gamma(\alpha)} \int_{t_k}^{t_{k+1}} (t_n - \tau)^{\alpha-1} \Big[H\|U(\tau) - U(t_{k+1})\| + K\|U(\tau) - U(t_k)\| \Big] \mathrm{d}\tau.$$

To bound the norms above we look at the expansion (4.1) to analyze the smoothness of the solution. In particular, we may fix a value, say t_p for a given $p > 0$, such that, in $[0, t_p]$ the exact solution $U(t)$ is not smooth, say $U(t) = \mathcal{O}(t^\alpha)$, while in $[t_p, T]$ a better regularity holds, say $U(t) = \mathcal{O}(t)$.

Then, if $k < p$ there exist two constants M_1 and M_2 such that $\|U(\tau) - U(t_{k+1})\| \leq M_1 h^\alpha$ and $\|U(\tau) - U(t_k)\| \leq M_2 h^\alpha$ for any $\tau \in [t_k, t_{k+1}]$.

Differently, in $[t_p, t_n]$ the exact solution $U(t)$ has more regularity and therefore classical results on the remainder of the polynomial interpolation can be used to infer the existence of positive constants M_3 and M_4 such that, for any $\tau \in [t_k, t_{k+1}]$ and $k > p$, it is $\|U(\tau) - U(t_{k+1})\| \leq M_3 h$ and $\|U(\tau) - U(t_k)\| \leq M_4 h$.

We may then split T_n as

$$T_n = T_{n,a} + T_{n,b}, \quad T_{n,a} = \sum_{k=0}^{p-1} I_{n,k}, \quad T_{n,b} = \sum_{k=p}^{n-1} I_{n,k}.$$

Therefore when $n \leq p$ one can observe that

$$\|T_n\| = \|T_{n,a}\| \leq \frac{M_1 H + M_2 K}{\Gamma(\alpha)} h^\alpha \int_{t_0}^{t_n} (t_n - \tau)^{\alpha-1} \mathrm{d}\tau = C_1 h^\alpha t_n^\alpha$$

where for shortness we put $C_1 = (M_1 H + M_2 K)/\Gamma(\alpha)$.

When $n > p$ we have

$$\begin{aligned}
\|T_n\| &\leq \|T_{n,a}\| + \|T_{n,b}\| \\
&\leq C_1 h^\alpha \int_0^{t_p} (t_n - \tau)^{\alpha-1} \mathrm{d}\tau + C_2 h \int_{t_p}^{t_n} (t_n - \tau)^{\alpha-1} \mathrm{d}\tau \\
&\leq C_1 C_3 h^\alpha \int_0^{t_p} t_n^{\alpha-1} \mathrm{d}\tau + C_2 h \int_{t_p}^{t_n} (t_n - \tau)^{\alpha-1} \mathrm{d}\tau \\
&\leq \tilde{C}_1 h^{\alpha+1} t_n^{\alpha-1} + \tilde{C}_2 h t_n^\alpha
\end{aligned}$$

where $C_2 = (M_3 H + M_4 K)/\Gamma(\alpha)$ and we exploited the inequality

$$(t_n - \tau)^{\alpha-1} \leq C_3 t_n^{\alpha-1}$$

holding for some positive constant C_3 when $\tau < t_n$. Hence, since p is fixed, we can introduce further positive constants C_4, C_5 and C_6 in order to write

$$\hat{T_n} \leq \begin{cases} C_4 h^\alpha t_n^\alpha & n = 0, 1, \dots p \\ C_5 h^{\alpha+1} t_n^{\alpha-1} + C_6 h t_n^\alpha & n = p+1, p+2, \dots N. \end{cases} \tag{4.3}$$

Therefore, away from the origin, namely for $t_n \in [t_p, T]$, and h sufficiently small, we can find two constants V_1 and V_2 such that

$$\hat{T_n} \leq V_1 h^{\alpha+1} t_n^{\alpha-1} + V_2 h t_n^\alpha;$$

this result, inserted into the expression (4.2), completes the proof by reason of Lemma 4.1. □

5. Kernel compression scheme

As seen before, to evaluate the exact solution of the FDE (2.2) we may refer to its integral formulation (3.1); however, it involves weakly-singular integrals whose computation may result seriously challenging since the presence of the singularity. In addition, most of the applications usually require an order α quite close to 1, thus implying that the kernel $t^{\alpha-1}$ decays slowly as $t \to +\infty$; as a consequence, all the past history of the solution, from 0 to t, must be taken into account to correctly evaluate the solution.

To separate these two features, we fix a value $\delta t > 0$ to rewrite (3.1) as

$$U(t) = U_0 + \underbrace{\frac{1}{\Gamma(\alpha)} \int_0^{t-\delta t} (t-\tau)^{\alpha-1} G(\tau, U(\tau)) \mathrm{d}\tau}_{U_{\mathrm{M}}(t)} + \underbrace{\frac{1}{\Gamma(\alpha)} \int_{t-\delta t}^{t} (t-\tau)^{\alpha-1} G(\tau, U(\tau)) \mathrm{d}\tau}_{U_{\mathrm{L}}(t)}$$

where $U_{\mathrm{M}}(t)$ is the memory term and $U_{\mathrm{L}}(t)$ is the local term. When δt is sufficiently small, the more demanding computation is related to the memory term $U_{\mathrm{M}}(t)$.

For the local term, any numerical method for weakly singular integral equations may be used. In particular, we apply the rectangular PI IMEX scheme described in Section 3 to evaluate $U_{\mathrm{L}}(t)$ at any grid-point t_n. Let us denote with $n^\star = \delta t/h$ (for simplicity δt is chosen such that $n^\star \in \mathbb{N}$) and we observe that the rectangular PI IMEX approximation (3.4) leads to

$$\begin{aligned} U_{\mathrm{L}}(t_n) &\approx \sum_{k=n-n^\star}^{n-1} \frac{1}{\Gamma(\alpha)} \int_{t_k}^{t_{k+1}} (t_n - \tau)^{\alpha-1} \big[L(U_{k+1}) + F(U_k)\big] \mathrm{d}\tau \\ &= h^\alpha \sum_{k=n-n^\star}^{n-1} b^{(\alpha)}_{n-k-1} L(U_{k+1}) + h^\alpha \sum_{k=n-n^\star}^{n-1} b^{(\alpha)}_{n-k-1} F(U_k) \\ &= h^\alpha \Big(b^{(\alpha)}_{n^\star-1} F(U_{n-n^\star}) + \sum_{k=n-n^\star+1}^{n-1} \big(b^{(\alpha)}_{n-k} L(U_k) + b^{(\alpha)}_{n-k-1} F(U_k) \big) + b^{(\alpha)}_0 L(U_n) \Big). \end{aligned}$$

5.1. **Fast evaluation of the memory term.**

To weaken computation and storage needs necessary to compute $U_{\mathrm{M}}(t)$ several strategies have been proposed. Here, we focus on the KCS [35; 43; 44; 45; 46; 47] which appears particularly suited for the problems under investigation. The basic idea of this approach is to use an accurate approximation of the integral representation for the kernel $t^{\alpha-1}/\Gamma(\alpha)$, namely

$$\frac{t^{\alpha-1}}{\Gamma(\alpha)} = \sum_{q=1}^{Q} w_q \mathrm{e}^{-\xi_q t} + v_\alpha(t), \quad \max_{t \in [\delta t, T]} \big| v_\alpha(t) \big| \le \epsilon \tag{5.1}$$

for a given tolerance $\epsilon > 0$ (we will give the theoretical foundation of it in Proposition 5.2).

The memory term can now be written as

$$
\begin{aligned}
U_{\mathrm{M}}(t) &= \frac{1}{\Gamma(\alpha)}\int_0^{t-\delta t}(t-\tau)^{\alpha-1}G(\tau,U(\tau))\mathrm{d}\tau \\
&= \sum_{q=1}^{Q} w_q \int_0^{t-\delta t} \mathrm{e}^{-\xi_q(t-\tau)}G(\tau,U(\tau))\mathrm{d}\tau + \int_0^{t-\delta t} v_\alpha(t-\tau)G(\tau,U(\tau))\mathrm{d}\tau \\
&= \sum_{q=1}^{Q} w_q \mathrm{e}^{-\xi_q\delta t}U_{\mathrm{M},q}(t) + R(t)
\end{aligned}
\tag{5.2}
$$

with

$$
U_{\mathrm{M},q}(t) = \int_0^{t-\delta t} \mathrm{e}^{-\xi_q(t-\delta t-\tau)}G(\tau,U(\tau))\mathrm{d}\tau = \int_{\delta t}^{t} \mathrm{e}^{-\xi_q(t-\tau)}G(\tau-\delta t,U(\tau-\delta t))\mathrm{d}\tau,
$$

and $R(t)$ the remainder which satisfies

$$
\big\|R(t)\big\| \le \int_0^{t-\delta t} \big|v_\alpha(t-\tau)\big|\cdot\big\|G(\tau,U(\tau))\big\|\mathrm{d}\tau \le \epsilon \int_0^{t-\delta t} \big\|G(\tau,U(\tau))\big\|\mathrm{d}\tau \le \epsilon t \max_{\tau\in[0,t]}\big\|G(\tau,U(\tau))\big\|
$$

It is easy to observe that $U_{\mathrm{M},q}(t)$ is the solution of the initial value problem

$$
\begin{cases} y'(t) = -\xi_q y(t) + G(t-\delta t, U(t-\delta t)), \\ y(\delta t) = 0 \end{cases}
$$

for $t \ge \delta t$ (observe that the expression above is not properly a delayed differential equation since $U(t)$ must be considered as a known external term). Each of these systems can be solved, at a reasonable cost, by means of the implicit Euler scheme (using an implicit method is mandatory since some of the ξ_q can be large and thus make the system stiff). So, if $U_{\mathrm{M},q}^{(n)}$ denotes the approximation of $U_{\mathrm{M},q}(t_n)$, since $\delta t = n^\star h$, we evaluate for $n > n^\star$

$$
U_{\mathrm{M},q}^{(n)} = \frac{1}{1+h\xi_q}\Big(U_{\mathrm{M},q}^{(n-1)} + hL(U_{n-n^\star}) + hF(U_{n-n^\star})\Big), \quad U_{\mathrm{M},q}^{(n^\star)} = 0.
$$

Then equation (5.2) reads as

$$
\begin{aligned}
U_{\mathrm{M}}(t_n) \approx U_{\mathrm{M}}^{(n)} &= \sum_{q=1}^{Q} w_q \mathrm{e}^{-\xi_q\delta t}U_{\mathrm{M},q}^{(n)} \\
&= \sum_{q=1}^{Q} \frac{w_q \mathrm{e}^{-\xi_q\delta t}}{1+h\xi_q}U_{\mathrm{M},q}^{(n-1)} + h\sum_{q=1}^{Q} \frac{w_q \mathrm{e}^{-\xi_q\delta t}}{1+h\xi_q}\Big(L(U_{n-n^\star}) + F(U_{n-n^\star})\Big).
\end{aligned}
$$

To recap, when combining the rectangular PI IMEX for the local term and the KCS for the memory term in the expression of U_n, namely $U_n = U_0 + U_{\mathrm{L}}^{(n)} + U_{\mathrm{M}}^{(n)}$, the numerical scheme that we propose reads as

$$
\begin{aligned}
U_n - \frac{h^\alpha}{\Gamma(\alpha+1)}L(U_n) = U_0 &+ \sum_{k=n-n^\star+1}^{n-1}\Big(b_{n-k}^{(\alpha)}L(U_k) + b_{n-k-1}^{(\alpha)}F(U_k)\Big) \\
&+ \sum_{q=1}^{Q} \frac{w_q \mathrm{e}^{-\xi_q\delta t}}{1+h\xi_q}U_{\mathrm{M},q}^{(n-1)} + h\gamma L(U_{n-n^\star}) + \Big(h^\alpha b_{n^\star-1}^{(\alpha)} + h\gamma\Big)F(U_{n-n^\star})
\end{aligned}
\tag{5.3}
$$

where $\gamma = \sum_{q=1}^{Q}(w_q \mathrm{e}^{-\xi_q \delta t})/(1+h\xi_q)$.

As discussed in Section 3, the approximation above requires the solution of a linear system of the form (3.5) when the vector form is used, or the solution of a Sylvester equation like (3.6) when the matrix form is preferred.

5.2. **Use of Gauss-Jacobi quadrature for the kernel approximation.** We now address the problem of finding a suitable approximation (5.1) of the kernel $t^{\alpha-1}/\Gamma(\alpha)$. To start we describe an appropriate integral formulation to write it.

Proposition 5.1. *Let $0 < \alpha < 1$. For any $t > 0$ it is*

$$\frac{t^{\alpha-1}}{\Gamma(\alpha)} = \frac{2\sin\alpha\pi}{\pi} I_\alpha(t),$$

where

$$I_\alpha(t) = \int_{-1}^{1} (1-s)^{\alpha-1}(1+s)^{-\alpha}\psi_t(s)\mathrm{d}s, \quad \psi_t(s) = \frac{\mathrm{e}^{-\frac{1+s}{1-s}t}}{1-s}. \tag{5.4}$$

Proof. We start from the integral representation

$$\frac{t^{\alpha-1}}{\Gamma(\alpha)} = \frac{\sin\alpha\pi}{\pi}\int_0^\infty \mathrm{e}^{-rt}\frac{1}{r^\alpha}\mathrm{d}r, \tag{5.5}$$

which holds whenever $0 < \alpha < 1$ since $t^{\alpha-1}$ is the inverse Laplace transform of $\Gamma(\alpha)s^{-\alpha}$. We thus recast the integral above from an infinite to a finite interval by the change of variable $r = z(s)$ by means of the conformal map

$$z(s) : [-1,1) \to [0,\infty) = \frac{1+s}{1-s};$$

the claim then readily follows. □

Remark 5.1. In case one is interested in simulating equations with superdiffusion, namely when $1 < \alpha < 2$, a similar approach can be followed but, instead of using (5.5), it is necessary to start from the alternative integral representation of the kernel

$$\frac{t^{\alpha-1}}{\Gamma(\alpha)} = \frac{\sin\alpha\pi}{\pi}\int_0^\infty \frac{\mathrm{e}^{-rt}-1}{r^\alpha}\mathrm{d}r, \quad 1 < \alpha < 2.$$

To approximate the integral $I_\alpha(t)$ in (5.4) a Gauss-Jacobi (GJ) quadrature rule is the natural choice since the integrand contains the weight function

$$w(s) = (1-s)^{\alpha-1}(1+s)^{-\alpha}$$

which is typical of quadrature rules of GJ type; Jacobi polynomials are indeed orthogonal over $[-1,1]$ with respect to this weight function. GJ rules were already used in [48] for approximating matrix power functions A^α but with a different integral formulation.

In practice, for t very small or t very large, the function $\psi_t(s)$ presents a stiff behavior for values close to $s = -1$ or $s = 1$ respectively. This could lead to an accuracy loss in the numerical approximation of $I_\alpha(t)$. To make the integrand function in I_α smoother we use the equivalent expression holding for any parameter $c > 0$ (which must be suitably chosen)

$$\frac{t^{\alpha-1}}{\Gamma(\alpha)} = \frac{2c^{1-\alpha}\sin\alpha\pi}{\pi} I_\alpha(ct). \tag{5.6}$$

Regarding the related quadrature formula, we denote with $\tilde{\xi}_1, \dots, \tilde{\xi}_Q$ the Q roots of the Jacobi polynomial of degree Q and $\tilde{w}_1, \dots, \tilde{w}_Q$ the corresponding weights, so that the resulting quadrature formula for $I_\alpha(t)$ can be written as

$$I_\alpha^{[Q]}(t) = \sum_{q=1}^{Q} \tilde{w}_q \psi_t(\tilde{\xi}_q). \tag{5.7}$$

An accurate efficient method to compute weights and nodes, involving a computational cost only linear with respect to Q, has been recently proposed by Hale and Townsend [49].

Proposition 5.2. *Whenever $0 < \alpha < 1$ and $t > 0$ the following quadrature formula holds for any fixed positive parameter c*

$$\frac{t^{\alpha-1}}{\Gamma(\alpha)} = \sum_{q=1}^{Q} w_q \mathrm{e}^{-\xi_q t} + v_\alpha(ct) \tag{5.8}$$

with

$$w_q = \frac{2c^{1-\alpha}\tilde{w}_q \sin \alpha\pi}{\pi(1-\tilde{\xi}_q)} > 0, \quad \xi_q = c\frac{1+\tilde{\xi}_q}{1-\tilde{\xi}_q} > 0$$

and

$$v_\alpha(ct) = C_{Q,\alpha}\psi_{ct}^{(2Q)}(\eta)$$

where $\eta \in (-1,1)$ and

$$C_{Q,\alpha} = \frac{c^{1-\alpha}\Gamma(Q+\alpha)\Gamma(Q-\alpha+1)Q\sin\alpha\pi}{(2Q-1)!}\left[\frac{(Q-1)!}{\pi(2Q)!}\right]^2 2^{2Q+1}.$$

Proof. From (5.6) we deduce that

$$\frac{t^{\alpha-1}}{\Gamma(\alpha)} = \frac{2c^{1-\alpha}\sin\alpha\pi}{\pi}\big(I_\alpha^{[Q]}(t) + E\big).$$

Then, in the expression (5.7) for $I_\alpha^{[Q]}$ we make ψ_{ct} explicit and we collect terms to arrange weights and nodes as in the claim. The estimate of the quadrature error E hence follows from well-known results for GJ quadrature rules (e.g., see [50, Eq. (8.9.8)]). □

Coefficients $C_{Q,\alpha}$ in the quadrature error decay in a fast way. Indeed, by exploiting the results in [48, Lemma 3.2] we know that for any $0 < \alpha < 1$ it is

$$C_{Q,\alpha} \sim C_Q \coloneqq c^{1-\alpha}\sin\alpha\pi\frac{2^{-2Q+2}}{(2Q)!}, \quad Q \to \infty. \tag{5.9}$$

To give an estimate for the $2Q$-order derivatives of ψ_{ct} we present the following result.

Proposition 5.3. *Let $t > 0$, $c > 0$ and $Q \in \mathbb{N}$. Then for any $\rho > 1$ it is*

$$\max_{r\in(-1,1)} |\psi_{ct}^{(2Q)}(r)| \le (2Q)!\frac{\rho^2+1}{2\rho}\left[\frac{2\rho}{(\rho-1)^2}\right]^{2Q+2} \mathrm{e}^{\frac{(\rho+1)^2}{(\rho-1)^2}ct}. \tag{5.10}$$

Proof. To give an upper bound for $\psi_{ct}^{(2Q)}$ we consider for any $r \in (-1,1)$ the Cauchy integral formula

$$\psi_{ct}^{(2Q)}(r) = \frac{(2Q)!}{2\pi\mathrm{i}}\int_{\mathcal{C}} \frac{\psi_{ct}(z)}{(z-r)^{2Q+1}}\mathrm{d}z \tag{5.11}$$

where $\mathcal{C}$ is a closed contour with no singularities of $\psi_{ct}(z)$ on or within it. Since $\psi_{ct}(z)$ has just a singularity at $z = 1$, for convenience we choose $\mathcal{C} = E_{\rho,\epsilon} \cup L_- \cup C_{\sigma,\delta} \cup L_+$, as illustrated in Figure 1, and composed by the Bernstein ellipse

$$E_{\rho,\epsilon} \coloneqq \left\{ z \in \mathbb{C} \,\middle|\, z = g_\rho(\theta), \quad \epsilon \le \theta \le 2\pi - \epsilon \right\}, \quad g_\rho(\theta) = \frac{1}{2}\left(\rho \mathrm{e}^{\mathrm{i}\theta} + \frac{1}{\rho}\mathrm{e}^{-\mathrm{i}\theta} \right),$$

with a the small circular circuit enclosing $z = +1$

$$C_{\sigma,\delta} \coloneqq \left\{ z \in \mathbb{C} \,\middle|\, z = 1 + \sigma \mathrm{e}^{\mathrm{i}\theta}, \quad \delta \le \theta \le 2\pi - \delta \right\}, \quad \Im(\gamma_\epsilon) < \sigma < \Re(\gamma_\epsilon) - 1,$$

connected, in the proper sense, by two segments

$$L_- \coloneqq \left\{ z \in \mathbb{C} \,\middle|\, z = x - \mathrm{i}\Im(\gamma_\epsilon), \quad \Re(\gamma_\epsilon) \le x \le \Re(\gamma_\delta) \right\}$$
$$L_+ \coloneqq \left\{ z \in \mathbb{C} \,\middle|\, z = x + \mathrm{i}\Im(\gamma_\epsilon), \quad \Re(\gamma_\delta) \le x \le \Re(\gamma_\epsilon) \right\}$$

where, for shortness, we put $\gamma_\epsilon = g_\rho(\epsilon)$ and $\gamma_\delta = 1 + \sigma \mathrm{e}^{\mathrm{i}\delta}$, and δ is chosen such that $\Im(\gamma_\epsilon) = \Im(\gamma_\delta)$. Observe that

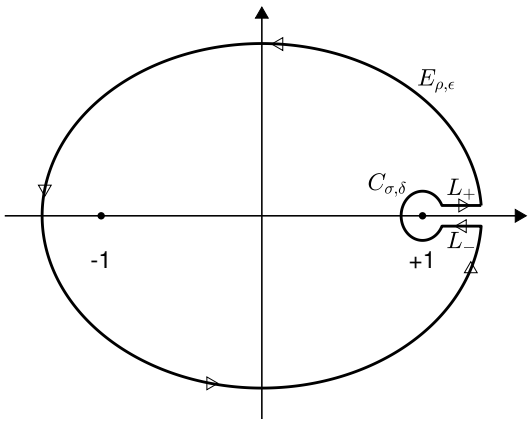

FIGURE 1. Contour for the Cauchy integral formula (5.11).

$$\int_{L_\pm} \frac{\psi_{ct}(z)}{(z-r)^{2Q+1}} \mathrm{d}z = \pm \int_{\Re(\gamma_\delta)}^{\Re(\gamma_\epsilon)} \frac{\psi_{ct}(x \pm \mathrm{i}\Im(\gamma_\epsilon))}{(x \pm \mathrm{i}\Im(\gamma_\epsilon) - r)^{2Q+1}} \mathrm{d}x$$

and when passing to the limit as $\epsilon \to 0$ (and hence $\delta \to 0$) we have

$$\lim_{\epsilon \to 0} \int_{L_\pm} \frac{\psi_{ct}(z)}{(z-r)^{2Q+1}} \mathrm{d}z = \pm \int_{1+\sigma}^{\frac{\rho^2+1}{2\rho}} \frac{\psi_{ct}(x)}{(x-r)^{2Q+1}} \mathrm{d}x$$

and hence the contributions on L_- and L_+ cancel each other out as $\epsilon \to 0$. On the small circuit $C_{\sigma,\delta}$

$$\int_{C_{\sigma,\delta}} \frac{\psi_{ct}(z)}{(z-r)^{2Q+1}} \mathrm{d}z = \int_{\delta}^{2\pi-\delta} \frac{\psi_{ct}(1+\sigma \mathrm{e}^{\mathrm{i}\theta})}{(1+\sigma \mathrm{e}^{\mathrm{i}\theta} - r)^{2Q+1}} \sigma \mathrm{i} \mathrm{e}^{\mathrm{i}\theta} \mathrm{d}\theta$$

and when we let the ray of the small circuit tend to zero, the contribution from this circuit tends to zero as well. Since the only contribution to (5.11) comes from the Bernstein ellipse, we observe that

$$\left| \psi_{ct}^{(2Q)}(r) \right| \le \frac{(2Q)!}{2\pi} \int_{\epsilon}^{2\pi-\epsilon} \left| \psi_{ct}(g_\rho(\theta)) \right| \left| \frac{g_\rho'(\theta)}{(g_\rho(\theta) - r)^{2Q+1}} \right| \mathrm{d}\theta$$

and elementary manipulations allow to verify that for any $r \in (-1,1)$ it is

$$\left|\psi_{ct}(g_\rho(\theta))\right| \le \frac{2\rho}{(\rho-1)^2}\mathrm{e}^{\frac{(\rho+1)^2}{(\rho-1)^2}ct}, \quad \left|g'_\rho(\theta)\right| \le \frac{\rho^2+1}{2\rho}, \quad \min_{\theta\in[0,2\pi]}|g_\rho(\theta)-r| \ge \frac{(\rho-1)^2}{2\rho}$$

from which the proof follows by letting again $\epsilon \to 0$. □

By using (5.9) and (5.10) in Proposition 5.2, we observe that for a sufficiently large number of nodes Q the quadrature error can be bounded by

$$|v_\alpha(ct)| \le 16c^{1-\alpha}\sin\alpha\pi\frac{\rho^2+1}{2\rho}\left[\frac{\rho}{(\rho-1)^2}\right]^{2Q+2}\mathrm{e}^{\frac{(\rho+1)^2}{(\rho-1)^2}ct}$$

and, when ρ is chosen sufficiently large so that $\rho/(\rho-1)^2 < 1$ (namely $\rho > 2+\sqrt{3}$) it is assured that any given accuracy can be achieved once a sufficiently large number of nodes is used.

The above estimate, however, can be quite conservative, especially for large t and not always turns to be useful to determine the number of nodes Q necessary to achieve a prescribed accuracy. For practical purposes we therefore propose an algorithmic approach to tune in a fine way the parameters Q and c in the quadrature rule.

5.3. **Fine tuning of quadrature parameters.** The accuracy of the time-step procedure (5.3), as based on the combination of PI, IMEX and KCS, depends on several factors. In particular the choice of the time step-size h will influence the accuracy of the PI IMEX method, whilst the accuracy of the KCS depends on the choice of the length interval δt on which to apply the PI IMEX and the number Q of nodes (together with the smoothing parameter c) in the quadrature rule (5.7).

All these parameters have influence not only on the accuracy of the solution but also on the computational cost. It is unfortunately not realistic to find the optimal choice of the parameters in order to achieve a certain accuracy with the lowest computational cost. We therefore propose an heuristic approach based on considerations from the nature of the problem and from the features of each method.

The step-size h and the interval length δt are fixed a priori, the former on the basis of the target accuracy for the solution and the latter according to computational needs (storage capability and execution time). The number Q of nodes and the smoothing parameter c in the quadrature rule (5.8) are suitably selected with the aim of preserving the $\mathcal{O}(h)$ accuracy of the PI IMEX. Therefore, it is fixed a tolerance $\varepsilon > 0$ and, by means of a direct search algorithm, the values Q are c are determined so that (5.1) holds on the selected interval $[\delta t, T]$.

As expected, (see Figures 2, 3 and 4), larger numbers Q of nodes are necessary to achieve the target tolerance when α or δt are smaller, as effect of the stronger singularity of the power kernel $t^{\alpha-1}$ close to the origin. Moreover, also the width of the integration interval plays a role as one can clearly observe from the comparison of the three considered upper end-points $T = 1$ (Figure 2), $T = 10$ (Figure 3) and $T = 20$ (Figure 4).

It is possible that the algorithm to determine the minimum number of nodes Q (and the smoothing parameter c) to achieve a target tolerance ε is not optimized and, surely, this point deserves a deeper investigation. However, the number of nodes Q provided by the algorithm appears suitable for a reasonably fast computation.

The tolerance ε must be chosen to preserve the $\mathcal{O}(h)$ accuracy of the PI IMEX. Since the accumulation of errors at each step, for integrating on an interval $[0,T]$, with $T = Nh$, a tolerance proportional to h/N appears necessary. Thus we choose

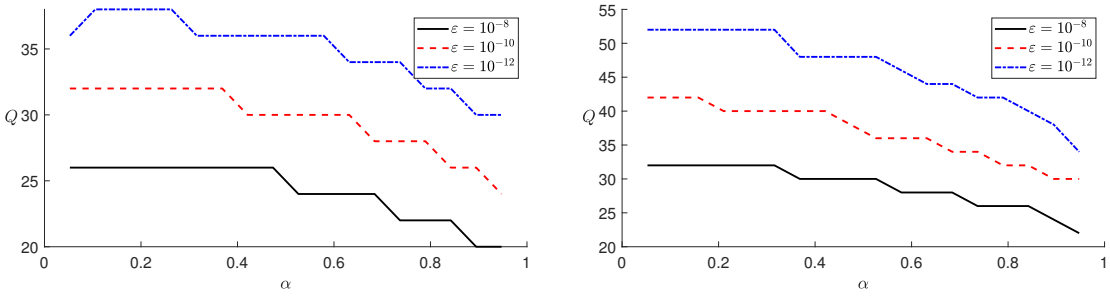

FIGURE 2. Number Q of nodes to achieve the target tolerance ε by the GJ quadrature (5.7) to approximate $t^{\alpha-1}/\Gamma(\alpha)$ in $[\delta t, T]$, when $T = 1$ and $\delta t = 0.01$ (left plot) or $\delta t = 0.005$ (right plot).

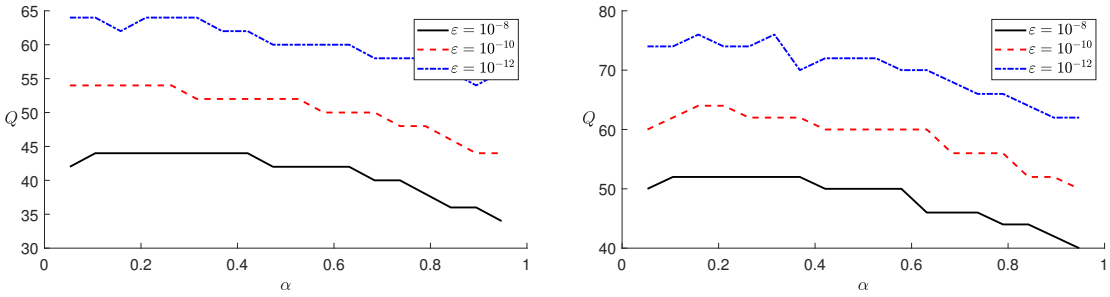

FIGURE 3. Number Q of nodes to achieve the target tolerance ε by the GJ quadrature (5.7) to approximate $t^{\alpha-1}/\Gamma(\alpha)$ in $[\delta t, T]$, when $T = 10$ and $\delta t = 0.01$ (left plot) or $\delta t = 0.005$ (right plot).

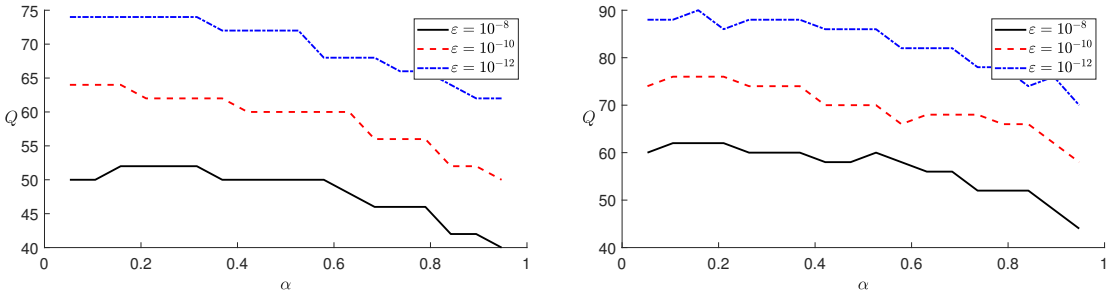

FIGURE 4. Number Q of nodes to achieve the target tolerance ε by the GJ quadrature (5.7) to approximate $t^{\alpha-1}/\Gamma(\alpha)$ in $[\delta t, T]$, when $T = 20$ and $\delta t = 0.01$ (left plot) or $\delta t = 0.005$ (right plot).

$\varepsilon = h^2/C$, where the factor C compensates for neglected constants. In our experiments we have observed that $C = 10$ is a sufficiently conservative value.

6. Numerical tests on the accuracy and efficiency of the time-step integration

We present here some numerical experiments in order to verify accuracy and computational efficiency of the proposed approach. All the experiments are carried out in Matlab ver. 9.9.91495850 (R2020b) on a PC equipped with an Intel i7-9600 CPU running at 3.0 GHz (with 16.0 Gbyte of RAM) under Windows 10 Pro.

With the only aim of testing the accuracy, we first consider a linear problem whose exact solution is known, in such a way as to calculate the actual error of the numerical

solution. In particular, we consider the problem

$$\begin{cases} {}^{C}\mathcal{D}_t^{\alpha} u = d\Delta u + f(u) & \\ \frac{\partial u}{\partial n} = 0 & (x,y) \in \partial\Omega \\ u\big|_{t=0} = u_0(x,y) & (x,y) \in \Omega \end{cases} \tag{6.1}$$

with $\Omega = [0,1] \times [0,1]$, $u_0(x,y) = \cos(\pi xy(1-x)(1-y))$ and $f(u) = -(dG_1(x,y) + c)u - dG_2(x,y)E_\alpha(-ct^\alpha)$. Here $E_\alpha(z)$ is the Mittag-Leffler function and G_1 and G_2 are two functions suitably selected in order to get the exact solution for this problem in the form $u(x,y,t) = u_0(x,y)E_\alpha(-ct^\alpha)$. The reference solution for $d = 0.4$, $c = 0.9$ and $\alpha = 0.8$ is presented in Figure 5.

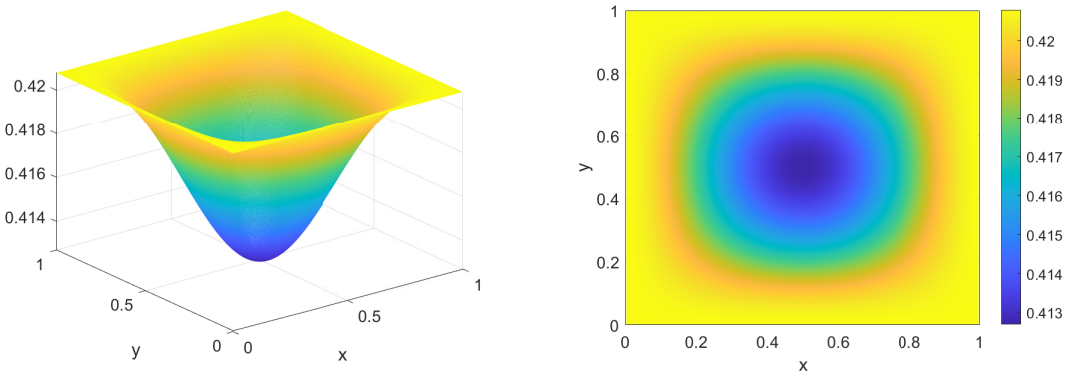

FIGURE 5. Exact solution at $T = 1$ of the test problem (6.1) with $\alpha = 0.8$.

Table 1 reports the relative errors, in $\|.\|_\infty$, together with the experimental order of convergence (EOC), of the numerical simulations for $\alpha = 0.7$, $\alpha = 0.8$ and $\alpha = 0.9$ when a space grid with $N_x = N_y = 400$ is considered. In Figure 6 these relative errors are plotted, together with the results for the same parameters and $N_x = N_y = 200$. As expected, the convergence is almost linear with respect to h.

h	$\alpha = 0.7$		$\alpha = 0.8$		$\alpha = 0.9$	
	Error	EOC	Error	EOC	Error	EOC
2^{-8}	1.061×10^{-3}		1.095×10^{-3}		1.036×10^{-3}	
2^{-9}	5.984×10^{-4}	0.826	6.655×10^{-4}	0.719	7.015×10^{-4}	0.562
2^{-10}	2.923×10^{-4}	1.034	3.341×10^{-4}	0.994	3.657×10^{-4}	0.940
2^{-11}	1.383×10^{-4}	1.079	1.612×10^{-4}	1.052	1.803×10^{-4}	1.020
2^{-12}	6.392×10^{-5}	1.114	7.574×10^{-5}	1.089	8.610×10^{-5}	1.066
2^{-13}	2.795×10^{-5}	1.194	3.381×10^{-5}	1.164	3.902×10^{-5}	1.142

TABLE 1. Relative errors in $\|\cdot\|_\infty$ at $T = 1$ for the test problem (6.1) when $N_x = N_y = 400$.

We now consider the Fischer-KPP equation describing the time evolution of the spread of the density of some populations or epidemics diffusing and reacting at the same time [51]. For simplicity, we confine to a normalized unit square domain $\Omega = [0,1] \times [0,1]$ and, as usual with models of this type, it is assumed that the system is isolated from outside the domain thus to impose homogeneous Neumann boundary conditions

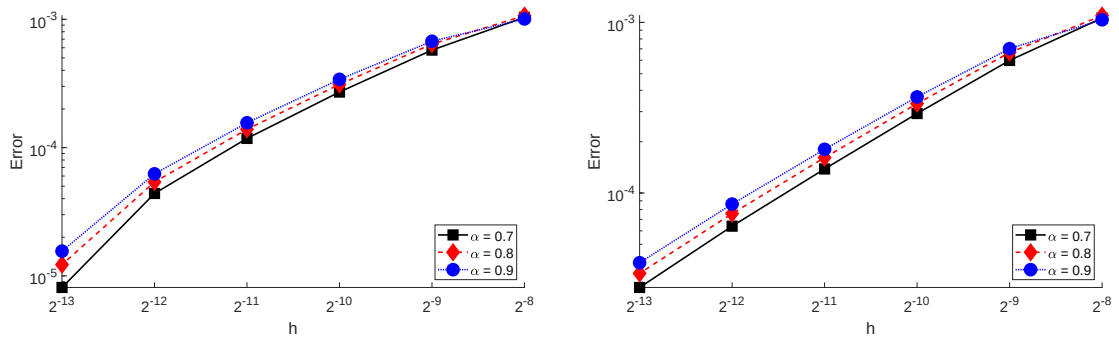

FIGURE 6. Relative errors at $T = 1$ of the test problem (6.1) when $N_x = N_y = 200$ (left plot) and $N_x = N_y = 400$ (right plot).

$$
(6.2) \qquad \begin{cases} {}^{\mathrm{C}}\mathcal{D}_t^{\alpha} u = d\Delta u + ru(1-u) & \\ \frac{\partial u}{\partial n} = 0 & (x,y) \in \partial\Omega \\ u\big|_{t=0} = x^2(1-x)^2 y^2 (1-y)^2 & (x,y) \in \Omega \end{cases}
$$

Values $d = 0.4$ and $r = 0.9$ for the parameters of the equation are considered. Since an exact solution is not available for this equation, we will adopt as reference solution the numerical approximation obtained with a very small step-size (namely $h = 2^{-18} \approx 2.8 \times 10^{-6}$). This reference solution for $\alpha = 0.8$ is illustrated in Figure 7.

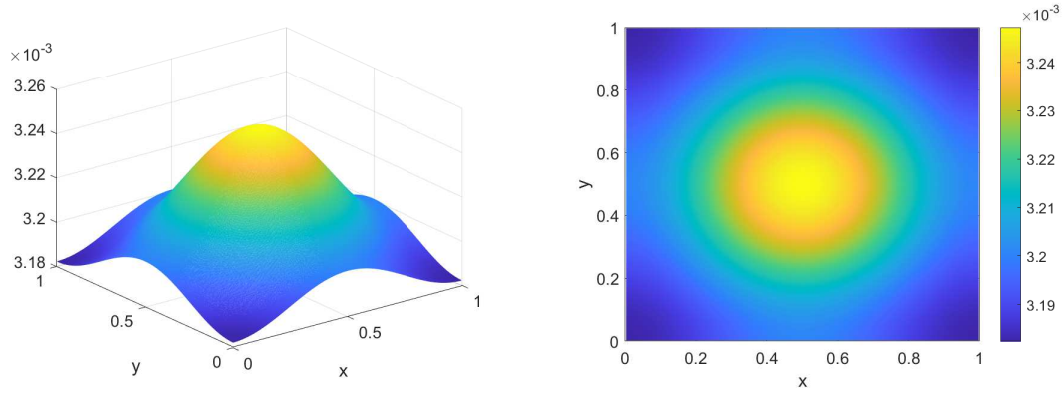

FIGURE 7. Exact solution at $T = 1$ of the test problem (6.2) with $\alpha = 0.8, d = 0.4$ and $r = 0.9$.

Table 2 reports the relative errors, in $\|.\|_\infty$, of the numerical simulations with respect to the aforementioned reference solutions. Results refer to $\alpha = 0.7$, $\alpha = 0.8$ and $\alpha = 0.9$ when a space grid with $N_x = N_y = 400$ is considered. Also in this case, the first order for convergence is achieved.

The same information can be inferred from the plots of the relative errors presented in Figure 8 where the errors for $N_x = N_y = 200$ (left plot) are shown together with those for $N_x = N_y = 400$ (right plot):

We now focus on the advantages of the proposed approach (5.3) from the computational point of view. For this purpose we perform a comparison with different strategies. We consider the Fisher problem (6.2) on the interval $[0, T]$ and we fix a step–size $h > 0$, such that $N_t = T/h$ grid-points are involved. The following approaches will be compared:

h	$\alpha = 0.7$		$\alpha = 0.8$		$\alpha = 0.9$	
	Error	EOC	Error	EOC	Error	EOC
2^{-11}	5.299×10^{-5}		6.706×10^{-5}		7.472×10^{-5}	
2^{-12}	3.755×10^{-5}	0.497	4.199×10^{-5}	0.675	4.397×10^{-5}	0.765
2^{-13}	2.170×10^{-5}	0.791	2.303×10^{-5}	0.866	2.344×10^{-5}	0.908
2^{-14}	1.139×10^{-5}	0.930	1.177×10^{-5}	0.969	1.180×10^{-5}	0.991
2^{-15}	5.552×10^{-6}	1.037	5.647×10^{-6}	1.059	5.616×10^{-6}	1.071

TABLE 2. Relative errors in $\|\cdot\|_\infty$ at $T = 1$ for the test problem (6.2) when $N_x = N_y = 400$.

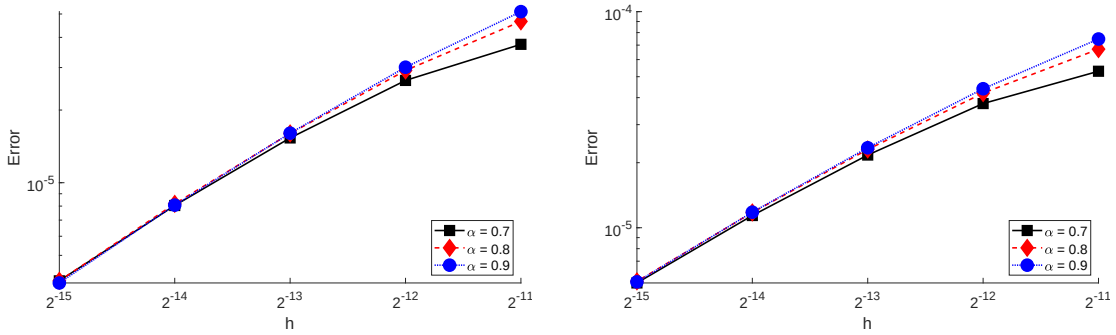

FIGURE 8. Relative errors at $T = 1$ of the test problem (6.2) when $N_x = N_y = 200$ (left plot) and $N_x = N_y = 400$ (right plot).

PI IMEX:: standard PI rule implemented in the IMEX way, as defined in (3.4); this method is expected to be extremely slow since the computation involves a number of floating-point operations proportional to $\mathcal{O}(N_t^2)$;

PI IMEX FFT:: in the PI IMEX method (3.4) the memory term is evaluated according to the FFT-based algorithm proposed in [29; 30] which provides a computational cost proportional to $\mathcal{O}(N_t(\log_2 N_t)^2)$ (see also [34] for a description of its implementation);

PI IMEX KCS VET:: this is the PI IMEX method with the KCS for the efficient treatment of the memory term described in (5.3) implemented in the standard vector formulation (2.1); its overall cost is expected to be proportional to $N_t Q$; the involved linear systems are solved by resorting to a LU decomposition.

PI IMEX KCS MAT:: this is the PI IMEX method with the KCS for the efficient treatment of the memory term described in (5.3) implemented in the matrix form (2.2).

The various plots in Figure 9 report the CPU time for each of the four approaches for the evaluation of the solution at each point in the time mesh; as we can see, the combination of PI IMEX with KCS turns out to be more competitive with respect to standard PI IMEX. However we observe that the representation of the spatial discretization by means of the matrix approach is more efficient when accurate discretizations of the spatial derivative are required (and hence large systems are involved). This feature confirms the suitability of the matrix approach for problems of large size also in the fractional-order case, as already observed in [4] for the integer-order case. Note that the CPU times of the different algorithms assume different values at $t = 0$ since each algorithm involves different initialization times which are taken into account in the time registration.

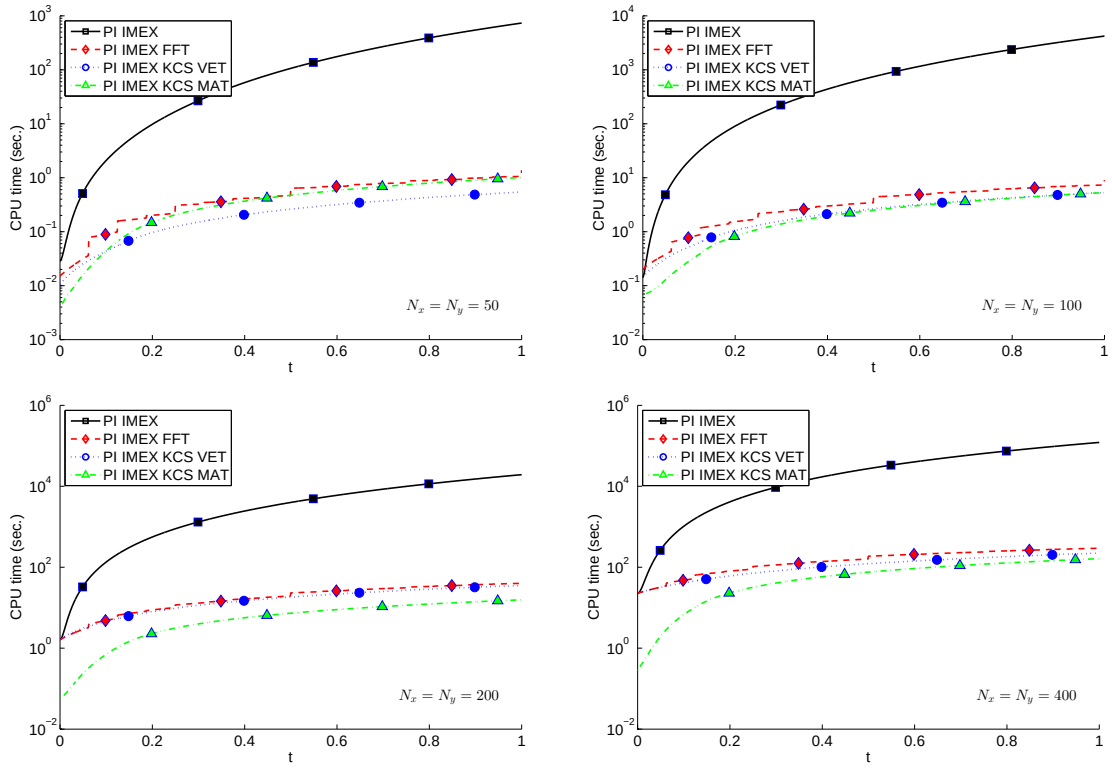

FIGURE 9. Comparison of CPU times for the test problem (6.2) with $\alpha = 0.8$, $h_t = 2^{-10}$ and $T = 1$ and different sizes of the spatial discretization.

To show the potentials and a possible field of applications of the proposed approach on long-time integration, we consider two time-fractional RD systems which present patterns, namely the fractional activator-inhibitor model and the fractional Fitzhugh-Nagumo model. Patterns usually emerge for systems characterized by two homogeneously distributed interacting substances and represent regional differences in the concentrations of the two substances.

We first consider a time-fractional version of the activator-inhibitor model [52] originally proposed to describe the concentrations of two chemical species: the activator u and the inhibitor v.

By following the notation successively adopted in [53], the subdiffusive activator-inhibitor system is defined by the coupled RD time-fractional equations

$$\begin{cases} {}^{\mathrm{C}}\mathcal{D}_t^{\alpha} u = \Delta u + u - av + buv - u^3 \\ {}^{\mathrm{C}}\mathcal{D}_t^{\alpha} v = d\Delta v + u - cv \\ \frac{\partial u}{\partial n} = 0 & (x,y) \in \partial\Omega \\ u\big|_{t=0} = \sin(xy), \quad v\big|_{t=0} = \cos(xy) & (x,y) \in \Omega \end{cases} \tag{6.3}$$

where a square domain $\Omega = [0, 50] \times [0, 50]$ is considered and the chosen parameters are $a = 7.81, b = 0.75, c = 5$ and $d = 20$.

In Figure 10 we present the results of the numerical simulation at $T = 1000$ with $h_t = 2^{-9}$, $N_x = N_y = 200$ and $\alpha = 0.95$. For a problem of this kind standard approaches would not have been able to provide any result in a reasonable time while our method reveals some patterns in the considered domain.

As a further example we consider the Fitzhugh-Nagumo equation, an important nonlinear reaction-diffusion equation applied to model the transmission of electrical

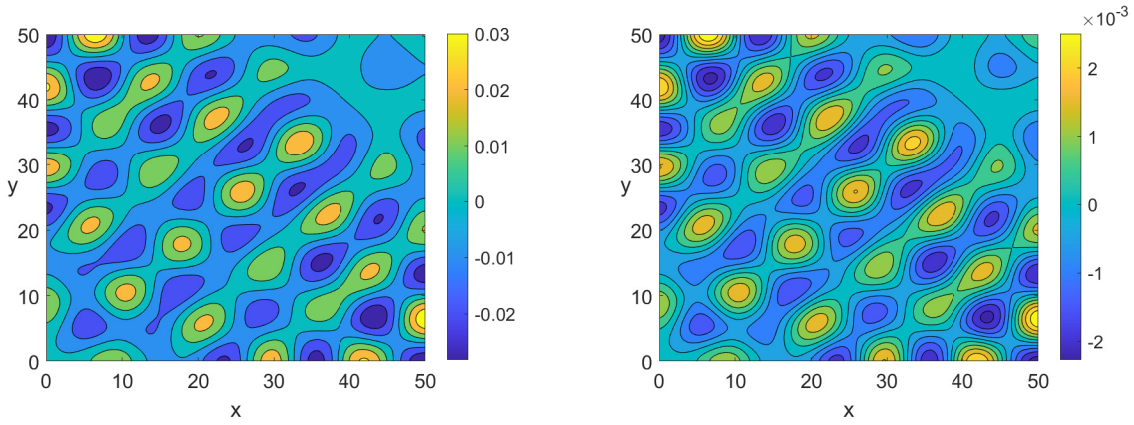

FIGURE 10. Numerical solution of Eq. (6.3) at $T = 1000$ with $h_t = 2^{-9}$, $N_x = N_y = 200$ and $\alpha = 0.95$. Solution u (left plot) and solution v (right plot).

impulses along a nerve fiber [54; 55] and, more recently, in the modeling of population dynamics. We consider here the model discussed by Gambino et al. [56] in which we replace the integer-order derivative with the fractional Caputo derivative

$$
\begin{cases}
{}^{\mathrm{C}}\mathcal{D}_t^{\alpha} u = \Delta u + c(-u^3 + u - v) \\
{}^{\mathrm{C}}\mathcal{D}_t^{\alpha} v = d\Delta v + bc(u - av) \\
\frac{\partial u}{\partial n} = 0 & (x, y) \in \partial\Omega \\
u\big|_{t=0} = u_0(x, y), \quad v\big|_{t=0} = v_0(x, y) & (x, y) \in \Omega
\end{cases} \tag{6.4}
$$

on the square domain $\Omega = [0, \pi] \times [0, \pi]$ with parameters $a = 0.1, b = 11, c = 65.731$ and $d = 42.1887$. The initial conditions are taken as small perturbations of the equilibria. Results for the numerical simulations are presented in Figure 11 where we can observe the appearance of square patterns, which are typical of this model. In this case, for ease of presentation we have plotted the difference between the obtained solutions and their mean values.

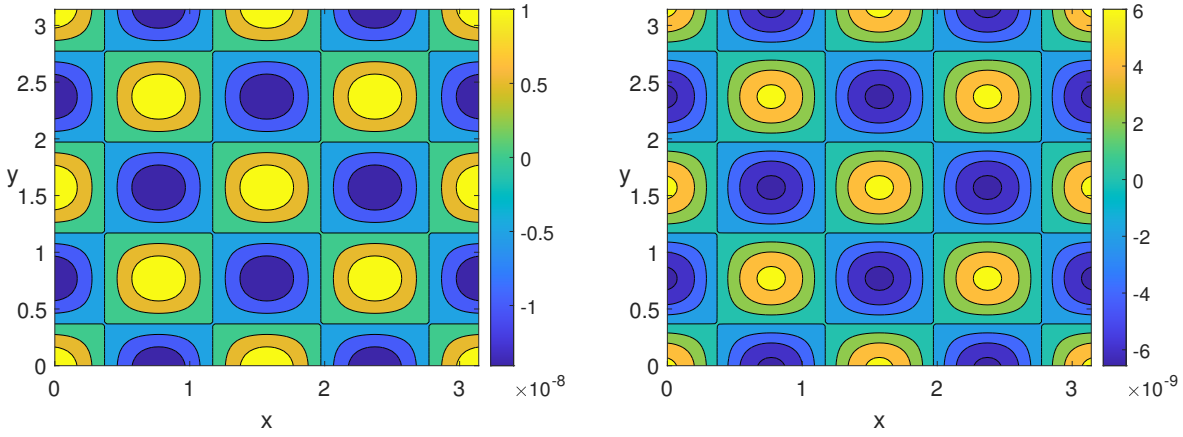

FIGURE 11. Numerical solution of Eq. (6.4) at $T = 200$ with $h_t = 2^{-10}$, $N_x = N_y = 100$ and $\alpha = 0.95$. Solution u (left plot) and solution v (right plot).

7. Conclusions

We have discussed a strategy to numerically solve time-fractional reaction diffusion problems: the main idea is the combination of product integration rules, in an implicit-explicit framework, with a kernel compression scheme to reduce the overload of computation and storage needs related to the persistent memory of fractional-order problems. Furthermore, at first we have discretized the problem with respect to the space variables and a matrix formulation of the semidiscretized problem, recently discussed for reaction-diffusion problems of integer order, has been used to improve the overall method. By a suitable selection of integration parameters of the KCS we have been able to preserve the original accuracy of the PI-IMEX scheme with a considerable reduction of the computational effort. The matrix formulation contributes to keep the computation at a reasonable level as well, and that turns to be extremely useful for integration over large intervals as in the case of applications in detection of pattern formation.

A series of numerical experiments has showed the efficiency of the proposed strategy and confirmed the theoretical results.

In a forthcoming research activity we aim to use more efficient strategies for the discretization of partial differential equations [57; 58] and to extend the analysis to the case of discontinuous systems [59; 60] which represent an interesting research topic with considerable applications in real life phenomena.

References

[1] W. Hundsdorfer, J. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations, 1st Edition, Springer Series in Computational Mathematics 33, Springer-Verlag Berlin Heidelberg, 2003.

[2] M. Stynes, Numerical methods for convection-diffusion problems or the 30 years war, arXiv preprint arXiv:1306.5172.

[3] D. Palitta, V. Simoncini, Matrix-equation-based strategies for convection–diffusion equations, BIT Numerical Mathematics 56 (2) (2016) 751–776.

[4] M. C. D'Autilia, I. Sgura, V. Simoncini, Matrix-oriented discretization methods for reaction–diffusion pdes: Comparisons and applications, Computers & Mathematics with Applications 79 (7) (2020) 2067–2085.

[5] J. D. Murray, Mathematical Biology II: Spatial Models and Biomedical Applications, Vol. 18 of Interdisciplinary Applied Mathematics, Springer New York, 2003.

[6] A. M. Turing, The chemical basis of morphogenesis, Phil. Trans. R. Soc. B (1952) 37–72.

[7] R. Caponetto, G. D. Dongola, L. Fortuna, I. Petráš, Fractional Order Systems: Modeling and Control Applications, Vol. 72 of Series on Nonlinear Science, Series A, World Scientific Publishing Co., 2010.

[8] A. Giusti, I. Colombaro, R. Garra, R. Garrappa, F. Polito, M. Popolizio, F. Mainardi, A practical guide to Prabhakar fractional calculus, Fract. Calc. Appl. Anal. 23 (1) (2020) 9–54.

[9] R. Herrmann, Fractional calculus: An introduction for physicists, World Scientific Publishing Co., Hackensack, NJ, 2018.

[10] A. A. Kilbas, H. M. Srivastava, J. J. Trujillo, Theory and applications of fractional differential equations, Vol. 204 of North-Holland Mathematics Studies, Elsevier Science B.V., Amsterdam, 2006.

[11] F. Mainardi, Fractional calculus and waves in linear viscoelasticity, Imperial College Press, London, 2010.
[12] V. E. Tarasov, Fractional dynamics, Nonlinear Physical Science, Springer, Heidelberg; Higher Education Press, Beijing, 2010.
[13] B. J. West, Nature's patterns and the fractional calculus, Vol. 2 of Fractional Calculus in Applied Sciences and Engineering, De Gruyter, Berlin, 2017.
[14] A. Jannelli, M. Ruggieri, M. P. Speciale, Analytical and numerical solutions of time and space fractional advection-diffusion-reaction equation, Commun. Nonlinear Sci. Numer. Simul. 70 (2019) 89–101.
[15] A. Jannelli, M. Ruggieri, M. P. Speciale, Numerical solutions of space-fractional advection-diffusion equations with nonlinear source term, Appl. Numer. Math. 155 (2020) 93–102.
[16] R. Metzler, W. G. Glöckle, T. F. Nonnenmacher, Fractional model equation for anomalous diffusion, Physica A: Statistical Mechanics and its Applications 211 (1) (1994) 13–24.
[17] B. I. Henry, T. A. M. Langlands, S. L. Wearne, Turing pattern formation in fractional activator-inhibitor systems, Phys. Rev. E 72 (2005) 026101.
[18] B. I. Henry, S. L. Wearne, Fractional reaction–diffusion, Physica A: Statistical Mechanics and its Applications 276 (3) (2000) 448 – 455.
[19] K. Diethelm, The analysis of fractional differential equations, Vol. 2004 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 2010.
[20] R. Garrappa, E. Kaslik, M. Popolizio, Evaluation of fractional integrals and derivatives of elementary functions: Overview and tutorial, Mathematics 7 (5) (2019) 407.
[21] I. Podlubny, Fractional differential equations, Vol. 198 of Mathematics in Science and Engineering, Academic Press Inc., San Diego, CA, 1999.
[22] G. M. Coclite, A. Fanizzi, L. Lopez, F. Maddalena, S. F. Pellegrino, Numerical methods for the nonlocal wave equation of the peridynamics, Appl. Numer. Math. 155 (2020) 119–139.
[23] M. D'Elia, Q. Du, C. Glusa, M. Gunzburger, X. Tian, Z. Zhou, Numerical methods for nonlocal and fractional models, Acta Numer. 29 (2020) 1–124.
[24] L. Lopez, S. Pellegrino, A spectral method with volume penalization for a nonlinear peridynamic model, International Journal for Numerical Methods in Engineering 122 (3) (2021) 707–725.
[25] L. Lopez, S. Pellegrino, A space-time discretization of a nonlinear peridynamic model on a 2D lamina (2021). arXiv:2102.06485.
[26] J. Henning, D. Palitta, V. Simoncini, K. Urban, Matrix oriented reduction of space-time Petrov-Galerkin variational problems, arXiv preprint arXiv:1912.10082.
[27] G. Kirsten, V. Simoncini, A matrix-oriented POD-DEIM algorithm applied to nonlinear differential matrix equations, arXiv preprint arXiv:2006.13289.
[28] Y. Hao, V. Simoncini, Matrix equation solving of PDEs in polygonal domains using conformal mappings (2020). doi:https://doi.org/10.1515/jnma-2020-0035.
[29] E. Hairer, C. Lubich, M. Schlichte, Fast numerical solution of nonlinear Volterra convolution equations, SIAM J. Sci. Statist. Comput. 6 (3) (1985) 532–541.
[30] E. Hairer, C. Lubich, M. Schlichte, Fast numerical solution of weakly singular Volterra integral equations, J. Comput. Appl. Math. 23 (1) (1988) 87–98.
[31] A. Young, Approximate product-integration, Proc. Roy. Soc. London Ser. A. 224 (1954) 552–561.

[32] K. Diethelm, N. J. Ford, A. D. Freed, A predictor-corrector approach for the numerical solution of fractional differential equations, Nonlinear Dynam. 29 (1-4) (2002) 3–22.
[33] K. Diethelm, N. J. Ford, A. D. Freed, Detailed error analysis for a fractional Adams method, Numer. Algorithms 36 (1) (2004) 31–52.
[34] R. Garrappa, Numerical solution of fractional differential equations: A survey and a software tutorial, Mathematics 6 (2) (2018) 16.
[35] K. Diethelm, R. Garrappa, M. Stynes, Good (and not so good) practices in computational methods for fractional calculus, Mathematics 8 (3) (2020) 324.
[36] C. Lubich, Runge-Kutta theory for Volterra and Abel integral equations of the second kind, Math. Comp. 41 (163) (1983) 87–102.
[37] M. Stynes, Too much regularity may force too much uniqueness, Fract. Calc. Appl. Anal. 19 (2016) 1554–1562.
[38] M. Stynes, Singularities, in: Handbook of fractional calculus with applications. Vol. 3, De Gruyter, Berlin, 2019, pp. 287–305.
[39] J. Dixon, On the order of the error in discretization methods for weakly singular second kind Volterra integral equations with nonsmooth solutions, BIT 25 (4) (1985) 624–634.
[40] R. Garrappa, On linear stability of predictor-corrector algorithms for fractional differential equations, Int. J. Comput. Math. 87 (10) (2010) 2281–2290.
[41] V. Simoncini, Computational methods for linear matrix equations, SIAM Review 58 (3) (2016) 377–441.
[42] T. Breiten, V. Simoncini, M. Stoll, Low-rank solvers for fractional differential equations, Electronic Transactions on Numerical Analysis 45 (2016) 107–132.
[43] J.-R. Li, A fast time stepping method for evaluating fractional integrals, SIAM J. Scientific Computing 31 (2010) 4696–4714.
[44] D. Baffet, J. S. Hesthaven, A kernel compression scheme for fractional differential equations, SIAM Journal on Numerical Analysis 55 (2) (2017) 496–520.
[45] D. Baffet, J. S. Hesthaven, High-order accurate adaptive kernel compression time-stepping schemes for fractional differential equations, Journal of Scientific Computing 72 (3) (2017) 1169–1195.
[46] D. Baffet, A Gauss–Jacobi kernel compression scheme for fractional differential equations, Journal of Scientific Computing 79 (1) (2019) 227–248.
[47] L. Banjai, M. López-Fernández, Efficient high order algorithms for fractional integrals and fractional differential equations, Numerische Mathematik 141 (2) (2019) 289–317.
[48] L. Aceto, C. Magherini, P. Novati, On the construction and properties of m-step methods for FDEs, SIAM J. Sci. Comput. 37 (2) (2015) A653–A675.
[49] N. Hale, A. Townsend, Fast and accurate computation of Gauss–Legendre and Gauss–Jacobi quadrature nodes and weights, SIAM Journal on Scientific Computing 35 (2) (2013) A652–A674.
[50] F. B. Hildebrand, Introduction to numerical analysis, 2nd Edition, Dover Publications, Inc., New York, 1987.
[51] J. Roessler, H. Hüssner, Numerical solution of the $(1 + 2)$-dimensional Fisher's equation by finite elements and the Galerkin method, Math. Comput. Modelling 25 (3) (1997) 57–67.
[52] V. Gafiychuk, B. Datsko, V. Meleshko, Mathematical modeling of time fractional reaction–diffusion systems, Journal of Computational and Applied Mathematics 220 (1) (2008) 215 – 225.

[53] L. Zhang, C. Tian, Turing pattern dynamics in an activator-inhibitor system with superdiffusion, Phys. Rev. E 90 (2014) 062915.
[54] R. FitzHugh, Impulses and physiological states in theoretical models of nerve membrane, Biophysical journal 1 (6) (1961) 445.
[55] J. Nagumo, S. Arimoto, S. Yoshizawa, An active pulse transmission line simulating nerve axon, Proceedings of the IRE 50 (10) (1962) 2061–2070.
[56] G. Gambino, M. Lombardo, G. Rubino, M. Sammartino, Pattern selection in the 2D FitzHugh–Nagumo model, Ricerche di Matematica 68 (2) (2019) 535–549.
[57] L. Lopez, G. Vacca, Spectral properties and conservation laws in mimetic finite difference methods for PDEs, J. Comput. Appl. Math. 292 (2016) 760–784.
[58] L. Beirão da Veiga, L. Lopez, G. Vacca, Mimetic finite difference methods for Hamiltonian wave equations in 2D, Comput. Math. Appl. 74 (5) (2017) 1123–1141.
[59] A. Colombo, N. Del Buono, L. Lopez, A. Pugliese, Computational techniques to locate crossing/sliding regions and their sets of attraction in non-smooth dynamical systems, Discrete Contin. Dyn. Syst. Ser. B 23 (7) (2018) 2911–2934.
[60] L. Lopez, S. Maset, Time-transformations for the event location in discontinuous ODEs, Math. Comp. 87 (313) (2018) 2321–2341.

Roberto Garrappa: Department of Mathematics, University of Bari, Via E. Orabona 4, 70125 Bari, Italy. Member of the INdAM Research group GNCS
Email address: roberto.garrappa@uniba.it
URL: www.dm.uniba.it/members/garrappa/main

Marina Popolizio: Department of Electrical and Information Engineering, Polytechnic University of Bari, Via E. Orabona 4, 70125 Bari, Italy. Member of the INdAM Research group GNCS
Email address: marina.popolizio@poliba.it