



Article

# A Collocation Method Based on Discrete Spline Quasi-Interpolatory Operators for the Solution of Time Fractional Differential Equations

Enza Pellegrino <sup>1,†</sup> , Laura Pezza <sup>2,‡</sup> and Francesca Pitolli <sup>2,\*</sup> <sup>1</sup> Department IIIIE, University of L'Aquila, 67100 L'Aquila, Italy; enza.pellegrino@univaq.it<sup>2</sup> Department SBAI, University of Roma "La Sapienza", 00161 Roma, Italy; laura.pezza@sbai.uniroma1.it

\* Correspondence: francesca.pitolli@sbai.uniroma1.it

† Current address: Department IIIIE, University of L'Aquila, Piazzale Ernesto Pontieri 2, Monteluco di Roio, 67100 L'Aquila, Italy.

‡ Current address: Department SBAI, University of Roma "La Sapienza", Via Antonio Scarpa 16, 00161 Roma, Italy.

**Abstract:** In many applications, real phenomena are modeled by differential problems having a time fractional derivative that depends on the history of the unknown function. For the numerical solution of time fractional differential equations, we propose a new method that combines spline quasi-interpolatory operators and collocation methods. We show that the method is convergent and reproduces polynomials of suitable degree. The numerical tests demonstrate the validity and applicability of the proposed method when used to solve linear time fractional differential equations.

**Keywords:** fractional differential equation; quasi-interpolatory operator; collocation method; B-spline



**Citation:** Pellegrino, E.; Pezza, L.; Pitolli, F. A Collocation Method Based on Discrete Spline Quasi-Interpolatory Operators for the Solution of Time Fractional Differential Equations. *Fractal Fract.* **2021**, *5*, 5. <https://doi.org/10.3390/fractalfract5010005>

Received: 28 November 2020

Accepted: 31 December 2020

Published: 5 January 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Fractional calculus refers to the calculus of integrals and derivatives of any real positive order [1–3]. During the last few years, it became very popular due to its applications in various fields of science and engineering. In fact, in many applications, real phenomena can be modeled by differential equations having a fractional derivative both in time and space. This allows one to take into account in a clear way the history and nonlocal behavior of the physical quantities under study. Through fractional calculus, we can study, for example, the viscoelasticity properties of special materials, anomalous diffusion in human tissues, water flow in deep ground, the prediction of earthquakes, and fractal problems [4–7].

There are several definitions of the fractional derivative that become equivalent under particular assumptions [1,3]. The first definition of a fractional derivative suitable for general functions appeared in the first half of the 19th Century and is now known as the Riemann–Liouville derivative:

$${}_{RL}D_t^\gamma y(t) := \frac{1}{\Gamma(m-\gamma)} \frac{d^m}{dt^m} \int_0^t \frac{y(\tau)}{(t-\tau)^{\gamma-m+1}} d\tau, \quad m-1 < \gamma < m, \quad m \in \mathbb{N}_+, \quad t > 0, \quad (1)$$

where:

$$\Gamma(\alpha) := \int_0^\infty \tau^{\alpha-1} e^{-\tau} d\tau$$

is the Euler gamma function and  $\mathbb{N}_+$  denotes the set of positive natural numbers.

Using the Riemann–Liouville definition, the usual differentiation operator in the Fourier domain can be easily extended to the fractional case, i.e.,

$$\mathcal{F}({}_{RL}D_t^\gamma y(t)) = (i\omega)^\gamma \mathcal{F}(y(t)), \quad \gamma \in \mathbb{R}_+, \quad (2)$$

where  $\mathcal{F}(\cdot)$  denotes the Fourier transform operator. Moreover, the Riemann–Liouville derivative coincides with the Grunwald–Letnikov derivative

$${}_{GL}D_t^\gamma y(t) = \lim_{\delta \rightarrow 0} \frac{1}{\delta^n} \sum_{k=0}^{\frac{t}{\delta}} (-1)^k \binom{\gamma}{k} y(t - \delta k), \quad (3)$$

where:

$$\binom{\gamma}{k} := \frac{\Gamma(\gamma + 1)}{k! \Gamma(\gamma - k + 1)}, \quad k \in \mathbb{N}, \quad \gamma \geq 0, \quad (4)$$

are the generalized binomial coefficients [1]. It is worth noticing that the sequence  $\{\binom{\gamma}{k}\}$  is compactly supported when  $\gamma \in \mathbb{N}$ , while for  $\gamma \notin \mathbb{N}$ , the sequence  $\{\binom{\gamma}{k}\}$  is no longer compactly supported, but its coefficients decay toward infinity as:

$$\binom{\gamma}{k} \rightarrow k^{-\gamma-1} \quad \text{for } k \rightarrow \infty.$$

The Grunwald–Letnikov definition (3) is easier to use when addressing the numerical solution of fractional differential equations by finite difference methods [8].

The Riemann–Liouville derivative is not suitable to be used in real-world models for two main reasons: the Riemann–Liouville derivative of constant functions is not zero, and the initial conditions for initial value fractional differential problems are not easy to enforce. To overcome these problems, in 1967, Caputo [9] introduced a new definition of the fractional derivative, now named the Caputo derivative:

$${}_{C}D_t^\gamma y(t) := \frac{1}{\Gamma(m - \gamma)} \int_0^t \frac{y^{(m)}(\tau)}{(t - \tau)^{\gamma - m + 1}} d\tau, \quad m - 1 < \gamma < m, \quad m \in \mathbb{N}_+, \quad t > 0. \quad (5)$$

The Riemann–Liouville derivative and the Caputo derivative are related to each other by:

$${}_{C}D_t^\gamma y(t) = {}_{RL}D_t^\gamma \left( y(t) - \sum_{\ell=0}^{m-1} \frac{t^\ell}{\ell!} y^{(\ell)}(0^+) \right) \quad (6)$$

so that they coincide when the function  $y(t)$  satisfies homogeneous initial conditions [1].

Even if the Caputo derivative requires higher smoothness of the function  $y(t)$ , it is more suitable to describe real phenomena since it shares some of the main properties with the ordinary derivative. In particular, the Caputo derivative of constant functions is zero, and the initial and boundary conditions are easy to implement. For this reason, in this paper, we consider an initial value differential problem having the Caputo derivative in time. For a discussion on other types of fractional derivatives, see [10].

In the literature, several numerical methods have been proposed to solve fractional differential equations [8,11,12]. Finite difference methods are very popular since they are easy to implement, but they require high order difference formulas to achieve high accuracy at the price of a high computational cost [13,14]. Spectral methods are more efficient since the solution to the differential problem is approximated by an expansion in a global polynomial basis, whose fractional derivative can be evaluated explicitly. However, the coefficients of the expansion are obtained by solving a dense linear system so that its numerical solution could have a high computational cost [11,14]. For this reason, expansions in local bases with small support are preferable. For instance, polynomial B-splines [15] are piecewise polynomials with small support that can be used to construct a basis for the spline space. In fact, spline functions are piecewise polynomials of a given degree and prescribed smoothness that can be constructed as linear combinations of B-splines. The freedom to choose the degree of the polynomial pieces and the overall smoothness of the spline give a great flexibility that can be used to adapt the spline to the function to be approximated.

As in the case of spectral methods, the expansion coefficients of the B-spline representation can be obtained using collocation methods with the advantage that the collocation

matrix is triangular and the off-diagonal entries decay fast. In the case of cardinal B-splines, i.e., B-splines with uniform knots, their fractional derivative can be evaluated explicitly by the backward finite difference operator [16]. This approach was introduced by two of the authors in [17] (see also [18]), where a spline collocation method based on B-spline bases was used to solve a time fractional diffusion equation. The method proved to be particularly effective also in the case of nonlinear fractional differential equations [19]. A different approach was considered in [20,21], where the fractional differential equation under study was reformulated as an integral equation and solved by approximating its solution by a spline function represented by a Lagrange polynomial basis. More recently, a similar approach was used in [22] to solve multi-term fractional differential equations. For a comprehensive treatment of this topic, see [23].

In this paper, we propose a new method based on refinable spline quasi-interpolatory operators. Quasi-interpolatory operators have greater flexibility with respect to interpolation, since they are only required to reproduce polynomials up to a given degree rather than interpolate a function at given points. For this reason, they can be constructed to satisfy some special properties, such as shape preserving properties or prescribed approximation order. In more detail, we consider an initial value problem having the fractional derivative in time and approximate its solution by a family of discrete quasi-interpolatory operators [24–26]. Then, the unknown coefficients of the approximating function are determined by the collocation method introduced in [18]. The method presented here is a generalization of the collocation method introduced in [18,27]. The main novelty is the use of spline quasi-interpolatory operators as approximating functions so that the unknown coefficients are suitable functionals of the solution to the differential problem. Different choices of the functionals allow us to increase the polynomial reproducibility of the approximating function with the advantage of achieving higher accuracy. To our knowledge, this is the first time that quasi-interpolatory operators have been used to numerically solve differential equations having a fractional derivative.

Since the quasi-interpolatory operators we consider are spline functions, i.e., piecewise polynomials of integer degree, their fractional derivatives are fractional splines, i.e., piecewise polynomials of non-integer degree [16]. Thus, the fractional derivatives needed to evaluate the numerical solution by the collocation method can be computed efficiently by an explicit differentiation rule [28]. The numerical results show that the method is easy to implement and efficient so that it can be used to solve other kinds of fractional differential problems, such as boundary value problems and nonlinear problems.

The paper is organized as follows. Some preliminaries, needed to develop the numerical method, are given in Section 2. In particular, in Section 2.1, the fractional differential equation we are interested in is introduced, as well as its analytical solution expressed in terms of the Mittag–Leffler function. In Section 2.2, the spline space we use as the approximating space is described. In Section 2.3, the family of discrete spline quasi-interpolatory operators we use as approximating functions is defined, and its main properties are analyzed. The new method we propose is introduced in Section 3 where the explicit expression of the fractional derivative of the B-spline basis is also given (cf. Section 3.1). Finally, in Section 4, some numerical tests are performed. Section 5 is devoted to the discussion of our results, while in Section 6, some conclusions are offered.

## 2. Materials and Methods

### 2.1. Linear Fractional Differential Problems

Let us consider the following initial value problem:

$$\begin{cases} {}_c D_t^\gamma y(t) + b y(t) = f(t), & t > 0, \quad 0 < \gamma < 1, \\ y(0) = y_0, \end{cases} \quad (7)$$

where  $b \in \mathbb{R}$  is a given parameter and  $f(t) \in C[0, +\infty)$  is the inhomogeneous term. The well-posedness, existence, and uniqueness of the solution to the fractional differential problem (7) were analyzed in [3] (§6). In particular, the continuity of  $f$  guarantees that there exists a unique solution  $y \in C[0, +\infty)$  that depends on a continuous way from the given data.

Using the Mittag–Leffler function [29]:

$$E_{\gamma,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\gamma k + \beta)}, \quad \gamma, \beta \geq 0, \quad z \in \mathbb{C},$$

the solution can be expressed in closed form as [3]:

$$y(t) = y_0 E_{\gamma,1}(-b t^\gamma) + \int_0^t f(t-\tau) \tau^{\gamma-1} E_{\gamma,\gamma}(-b \tau^\gamma) d\tau. \quad (8)$$

Our main purpose is to solve the differential problem (7) by a collocation method based on spline quasi-interpolatory operators. Even if the fractional differential problem here considered can be solved analytically, nevertheless its solution is a good test for analyzing the approximation properties of the proposed method and studying its convergence.

## 2.2. B-Spline Spaces

The cardinal B-spline  $B_n(t)$  is a piecewise polynomial of degree  $n$  having breakpoints on the integers. It can be defined through the backward finite difference operator, defined as:

$$\nabla^n f(t) := \sum_{k=0}^n \binom{n}{k} (-1)^k f(t-k),$$

where  $f$  is a continuous function, applied to the truncated power function:

$$T_n(t) := (\max(0, t))^n, \quad n \in \mathbb{N},$$

i.e.,

$$B_n(t) := \frac{1}{n!} \nabla^{n+1} T_n(t), \quad n \in \mathbb{N}. \quad (9)$$

The B-spline  $B_n(t)$  is compactly supported on  $I_n := [0, n+1]$ , positive in  $(0, n+1)$ , and belongs to  $C^{n-1}(\mathbb{R})$ . For further details on spline functions, see [15,30].

The system of the integer translates  $\mathcal{S}_n = \{B_n(t-k), k \in \mathbb{Z}\}$ , forming a basis for the space of spline functions of degree  $n$ . It reproduces polynomials up to degree  $n$ , i.e.,

$$t^{r-1} = \sum_{k \in \mathbb{Z}} \zeta_k^{(r)} B_n(t-k), \quad t \in \mathbb{R}, \quad 1 \leq r \leq n+1, \quad (10)$$

with:

$$\zeta_k^{(r)} = \frac{\text{symm}_{r-1}(k+1, \dots, k+n)}{\binom{n}{r-1}}, \quad k \in \mathbb{Z}, \quad (11)$$

where  $\text{symm}_r(t_1, \dots, t_p)$  denotes the symmetric function defined as:

$$\text{symm}_r(t_1, \dots, t_p) = \sum_{1 \leq k_1 < k_2 < \dots < k_r \leq p} t_{k_1} t_{k_2} \dots t_{k_r}.$$

(cf. [15]). In particular, for  $r = 1$ , it holds:

$$1 = \sum_{k \in \mathbb{Z}} B_n(t-k), \quad t \in \mathbb{R}, \quad n \in \mathbb{N}, \quad (12)$$

i.e., the function system  $\mathcal{S}_n$  forms a partition of unity, while for  $r = 2$ , we get:

$$t = \sum_{k \in \mathbb{Z}} \left( k + \frac{n+1}{2} \right) B_n(t-k), \quad t \in \mathbb{R}, \quad n \in \mathbb{N}_+.$$

The cardinal B-spline  $B_n(t)$  can be adapted to any set of equidistant breakpoints by dilation. In particular, choosing the dyadic nodes  $\{2^{-j}k, k \in \mathbb{Z}\}$  as breakpoints, we can define the refined basis:

$$\mathcal{S}_{nj} = \{B_{nj}(t), k \in \mathbb{Z}\}, \quad j \in \mathbb{Z}, \quad (13)$$

where the basis functions:

$$B_{nj}(t) = B_n(2^j t - k), \quad k \in \mathbb{Z}, \quad j \in \mathbb{Z}, \quad (14)$$

are the dilates and translates of  $B_n(t)$  having support on  $[2^{-j}k, 2^{-j}(k+n+1)]$ . Analogous to (10), for any  $j \in \mathbb{N}$ , one has:

$$t^{r-1} = \sum_{k \in \mathbb{Z}} \tilde{\zeta}_{jk}^{(r)} B_{nj}(t), \quad t \in \mathbb{R}, \quad 1 \leq r \leq n+1, \quad (15)$$

where the coefficients  $\{\tilde{\zeta}_{jk}^{(r)}\}$  are given by:

$$\tilde{\zeta}_{jk}^{(r)} = \frac{\text{symm}_{r-1}(2^{-j}(k+1), \dots, 2^{-j}(k+n))}{\binom{n}{r-1}}, \quad k \in \mathbb{Z}, \quad j \in \mathbb{N}. \quad (16)$$

### 2.3. Refinable Spline Quasi-Interpolatory Operators

Our interest is in the numerical solution of the fractional differential problem (7) using as the approximating function a quasi-interpolatory refinable operator constructed using the refinable basis  $\mathcal{S}_{nj}$ .

Let  $\mathcal{F}$  be a linear space of real-valued functions, and let  $\{\lambda_{ji}\}$  be a set of linear functionals  $\lambda_{ji} : \mathcal{F} \rightarrow \mathbb{R}$ . We assume that  $\mathcal{F}$  contains the class  $\mathbb{P}_r$  of polynomials of degree at most  $r-1$  with  $1 \leq r \leq n+1$ . Given a function  $f \in \mathcal{F}$ , a refinable quasi-interpolatory operator  $Q_{nj}f$  is an approximation to  $f$  of the form:

$$Q_{nj}f(t) = \sum_{k \in \mathbb{Z}} (\lambda_{jk}f) B_{nj}(t), \quad t \in \mathbb{R}, \quad j \in \mathbb{Z}, \quad (17)$$

such that  $Q_{nj}f$  is exact on polynomials up to degree  $r-1$ ,  $1 \leq r \leq n+1$ . We notice that, since any  $B_{nj}(t)$  has compact support, for any  $t \in \mathbb{R}$ , the sum in (17) is actually a finite sum.

It is easy to show that  $Q_{nj}f$  reproduces polynomials up to degree  $r-1$  if and only if:

$$\lambda_{jk} t^{\ell-1} = \tilde{\zeta}_{jk}^{(\ell)}, \quad 1 \leq \ell \leq r, \quad k \in \mathbb{Z}, \quad (18)$$

where  $\tilde{\zeta}_{jk}^{(\ell)}$  are the coefficients defined in (16) (cf. [25]). Since we will use the operator (17) to approximate the solution to the differential problem (7) in a finite discretization interval  $I = [0, T]$ ,  $T \in \mathbb{N}_+$ , in the following, we will consider the operator  $Q_{nj}f$  restricted to the interval  $I$ , i.e.,

$$Q_{nj}f(t) = \sum_{k=-n}^{N_j} (\lambda_{jk}f) B_{nj}(t), \quad t \in I, \quad j \geq j_0, \quad (19)$$

where  $N_j = 2^j T - 1$  and  $j_0 \geq \lceil \log_2((n+1)/T) \rceil$  is the starting refinement level.

To enforce (18), we express the functionals  $\lambda_{jk}f$  as:

$$\lambda_{jk}f = \sum_{i=1}^r \alpha_{jki} (\lambda_{jki}f), \quad 1 \leq r \leq n+1, \quad -n \leq k \leq N_j, \quad (20)$$

where  $\alpha_{jki}$  are real coefficients to be determined and  $\lambda_{jki}f : \mathcal{F} \rightarrow \mathbb{R}$  are linear functionals such that:

$$\det(\lambda_{jki}t^{\ell-1})_{1 \leq \ell, i \leq r} \neq 0. \quad (21)$$

There are several kinds of spline quasi-interpolatory operators [15,24,26,31–34]. A possible choice is to express the linear functionals  $\{\lambda_{jki}\}$  through divided difference operators. Let  $\{\tau_{jk1}, \dots, \tau_{jki}\}$  be a set of  $k$  distinct points in  $I \cap \text{supp}(B_{njk})$ . Then, we set:

$$\lambda_{jki}f = [\tau_{jk1}, \dots, \tau_{jki}]f, \quad 1 \leq i \leq r, \quad 1 \leq r \leq n+1, \quad (22)$$

i.e., the difference operator of order  $i-1$ . Therefore, we get the discrete quasi-interpolatory operator [24,26]:

$$dQ_{nj}f(t) = \sum_{k=-n}^{N_j} \sum_{i=1}^r \alpha_{jki} ([\tau_{jk1}, \dots, \tau_{jki}]f) B_{njk}(t). \quad (23)$$

**Proposition 1.** *The discrete quasi-interpolatory operator  $dQ_{nj}f$  reproduces polynomials up to degree  $r-1$  with  $1 \leq r \leq n+1$  if and only if:*

$$\sum_{i=1}^r \alpha_{jki} ([\tau_{jk1}, \dots, \tau_{jki}]t^{\ell-1}) = \zeta_{jk}^{(\ell)}, \quad 1 \leq \ell \leq r, \quad -n \leq k \leq N_j, \quad (24)$$

where  $\zeta_{jk}^{(\ell)}, 1 \leq \ell \leq n+1$  are given in (16).

**Proof.** The proof is similar to that in [35] (Corollary 1). For any  $k$  held fixed, (24) represents a linear system of  $r$  equations in the  $r$  unknowns  $\{\alpha_{jki}\}$ . Given the particular choice of the functionals  $\{\lambda_{jki}\}$  in (22), one has  $\det(\lambda_{jki}t^{r-1}) \neq 0, 1 \leq r \leq n+1, -n \leq k \leq N_j$ , so that the linear system has a unique solution.

Without loss of generality, we set  $r = n+1$  in (24). Then, inserting (24) into (23), we get:

$$dQ_{nj}t^{\ell-1} = \sum_{k=-n}^{N_j} \sum_{i=1}^{n+1} \alpha_{jki} [\tau_{jk1} \dots \tau_{jki}]t^{\ell-1} B_{njk}(t) = \sum_{i=-n}^{N_j} \zeta_{jk}^{(i)} B_{njk}(t) = t^{\ell-1}, \quad 1 \leq \ell \leq r+1,$$

i.e., the operator  $dQ_{nj}$  reproduces polynomials up to degree  $r-1$ . Oppositely, using (24) in (15), we get  $dQ_{nj}$  applied to monomials, and the claim follows.  $\square$

The coefficient matrix of the linear system (24) is lower triangular so that its solution can be obtained by the forward technique [24], i.e.,

$$\alpha_{jki} = \sum_{s=0}^{i-1} (-1)^s \zeta_{jk}^{(i-s)} \text{symm}_s(\tau_{jk1}, \dots, \tau_{jki-1}), \quad 1 \leq i \leq r, \quad -n \leq k \leq N_j. \quad (25)$$

In particular, from (25), it follows:

$$\alpha_{jk1} = 1, \quad \alpha_{jk2} = \zeta_{jk}^{(2)} - \tau_{jk1},$$

so that for  $r = 1$ , we get:

$$dQ_{nj}f(t) = \sum_{k=-n}^{N_j} f(\tau_{jk1}) B_{njk}(t), \quad (26)$$

which is the operator reproducing just the constant function considered in [28], while for  $r = 2$ , we get:

$$dQ_{nj}f(t) = \sum_{k=-n}^{N_j} \frac{f(\tau_{jk2})(\xi_{jk}^{(2)} - \tau_{jk1}) - f(\tau_{jk1})(\xi_{jk}^{(2)} - \tau_{jk2})}{\tau_{jk2} - \tau_{jk1}} B_{nj}(t). \quad (27)$$

### 3. Numerical Method

For any fixed  $j \geq j_0$ , we approximate the solution to the fractional differential problem (7) by the discrete quasi-interpolatory operator (23), i.e.,

$$y(t) \approx y_{nj}(t) = \sum_{k=-n}^{N_j} \sum_{i=1}^r \alpha_{jki}([\tau_{jk1}, \dots, \tau_{jki}]y) B_{nj}(t), \quad (28)$$

where  $\{\alpha_{jki}\}$  are the coefficients defined in (25) and:

$$y_{jki} = [\tau_{jk1} \dots \tau_{jki}]y, \quad 1 \leq i \leq r, \quad -n \leq k \leq N_j, \quad j \geq j_0,$$

are the unknown coefficients whose expression depends on the polynomial reproducibility order  $r$ . In particular, for  $r = 1$ :

$$y_{jki} = y(\tau_{jk1})$$

while for  $r = 2$ :

$$y_{jki} = \frac{y(\tau_{jk2})(\xi_{jk}^{(2)} - \tau_{jk1}) - y(\tau_{jk1})(\xi_{jk}^{(2)} - \tau_{jk2})}{\tau_{jk2} - \tau_{jk1}},$$

where  $\tau_{jki}$ ,  $-n \leq k \leq N_j$ ,  $i = 1, 2$  are given distinct points in the interval  $[2^{-j}k, 2^{-j}(k+n+1)] \cap I$ .

We determine the unknown coefficients by collocation. To this end, we choose a set of collocation points  $\{t_p, 0 \leq p \leq N_p\}$  that belong to the discretization interval  $I$  and enforce the approximating function  $y_{nj}(t)$  to solve the differential problem on these points, i.e.,

$$\begin{cases} {}_c D_t^\gamma y_{nj}(t_p) + b y_{nj}(t_p) = f(t_p), & 1 \leq p \leq N_s, \\ y_{nj}(0) = y_0. \end{cases} \quad (29)$$

Substituting (28) in the previous equations and rearranging the sums, we get:

$$\begin{cases} \sum_{k=-n}^{N_j} {}_c D_t^\gamma B_{nj}(t_p) \sum_{i=1}^r \alpha_{jki}(y_{jki}) + b \sum_{k=-n}^{N_j} B_{nj}(t_p) \sum_{i=1}^r \alpha_{jki}(y_{jki}) = f(t_p), & 1 \leq p \leq N_s, \\ \sum_{k=-n}^{N_j} B_{nj}(0) \sum_{i=1}^r \alpha_{jki}(y_{jki}) = y_0. \end{cases} \quad (30)$$

Equations (30) form a linear system that can be written in matrix form as:

$$\begin{cases} (D_{njs} + b B_{njs}) A_{jr} C_{jr} = F_s, \\ V_{njs} A_{jr} C_{jr} = y_0. \end{cases} \quad (31)$$

where  $C_{jr} = [y_{jki}, 1 \leq i \leq r, -n \leq k \leq N_j]^T$  is the unknown column vector,

$$A_{jr} = [\text{diag}(a_{j\ell}), 1 \leq \ell \leq r] \in \mathbb{R}^{(N_j+n+1) \times (N_j+n+1)r}, \quad a_{j\ell} = [\alpha_{j\ell k}, -n \leq k \leq N_j],$$

is the matrix collecting the coefficients  $\{\alpha_{jki}\}$ ,  $V_{njs} = [B_{njk}(0), -n \leq k \leq N_j]$  is the row vector whose entries are the basis functions evaluated at the initial point  $t = 0$ ,

$$D_{njs} = [{}_C D_t^\gamma B_{njk}(t_p), 1 \leq p \leq N_s, -n \leq k \leq N_j] \in \mathbb{R}^{N_s \times (N_j + n + 1)},$$

$$B_{njs} = [B_{njk}(t_p), 1 \leq p \leq N_s, -n \leq k \leq N_j] \in \mathbb{R}^{N_s \times (N_j + n + 1)},$$

are the collocation matrices of the refinable basis, and:

$$F_s = [f(t_p), 1 \leq p \leq N_s]^T,$$

is the known term.

The linear system (31) has  $N_s + 1$  equations and  $(N_j + n + 1)r$  unknowns. To guarantee the existence of a unique solution, for any fixed  $n$  and  $r$ , the number of collocation points  $N_s + 1$  and the refinement level  $j$  have to be chosen so that number of unknowns is no greater than the number of equations (cf. [27,36]). In the case when  $N_s + 1 > (N_j + n + 1)r$ , (31) results in an overdetermined linear system that can be solved by the least squares method. In this case, particular attention should be paid to fulfill the initial condition.

We notice that from the partition of unity property (12) and [18] (Th. 3.1), it follows that the collocation solution  $y_{nj}(t)$  is stable, i.e.,

$$\|y_{nj}(t)\|_\infty \leq \kappa \|f\|_\infty,$$

where  $\kappa$  is a constant independent of  $j$ .

Finally, the collocation method can be proven to be convergent (cf. [27,28]).

**Theorem 1.** *The collocation method is convergent, i.e.:*

$$\lim_{j \rightarrow \infty} \|y(t) - y_{nj}(t)\|_\infty = 0,$$

with convergence order  $\nu$  provided that  $y \in C^\nu[0, T]$ , i.e.,

$$\|y(t) - y_{nj}(t)\|_\infty = O(2^{-j\nu}).$$

**Proof.** Using Definition (5), it is easy to show that the fractional differential equation (7) is equivalent to the integral equation:

$$z(t) + \frac{b}{\Gamma(\gamma)} \int_0^t \frac{z(\tau)}{(t-\tau)^{1-\gamma}} d\tau = f(t),$$

where  $z(t) = {}_C D_t^\gamma y(t)$ . The collocation method can be used also to approximate the solution to this integral equation, i.e.,

$$z_{nj}(t_p) + \frac{b}{\Gamma(\gamma)} \int_0^{t_p} \frac{z_{nj}(\tau)}{(t_p - \tau)^{1-\gamma}} d\tau = f(t_p), \quad 1 \leq p \leq N_s,$$

where  $z_{nj}(t)$  is the quasi-interpolatory operator (23). Thus, the equivalence implies that the approximation error  $\|y(t) - y_{nj}(t)\|_\infty$  is the same as the approximation error  $\|z(t) - z_{nj}(t)\|_\infty$  (cf. [36]). Since  $z_{nj}$  is a spline operator, the convergence is guaranteed with approximation order  $\nu$  when  $y \in C^\nu[0, T]$  [30].  $\square$



### 3.1. The Fractional Derivative of the Refinable B-Spline Basis

To numerically solve the linear system (31), we need to evaluate the Caputo derivative of the basis functions  $B_{nj_k}$ ,  $-n \leq k \leq N_j$ . For  $0 \leq k \leq N_j$ , the Caputo derivative of  $B_{nj_k}(t)$  can be explicitly evaluated by [18,27]:

$${}_C D_t^\gamma B_{nj_k}(t) = 2^{\gamma j} \frac{\Delta^{n+1} T_{n-\gamma}(2^j t - k)}{\Gamma(n - \gamma + 1)}, \quad 0 < \gamma \leq n. \quad (32)$$

For  $-n \leq k \leq -1$ , the functions  $B_{nj_k}$  are left edge functions having support on  $[0, 2^{-j}(n+k+1)]$ . When  $0 < \gamma < 1$ , their fractional derivative can be explicitly evaluated by [27]:

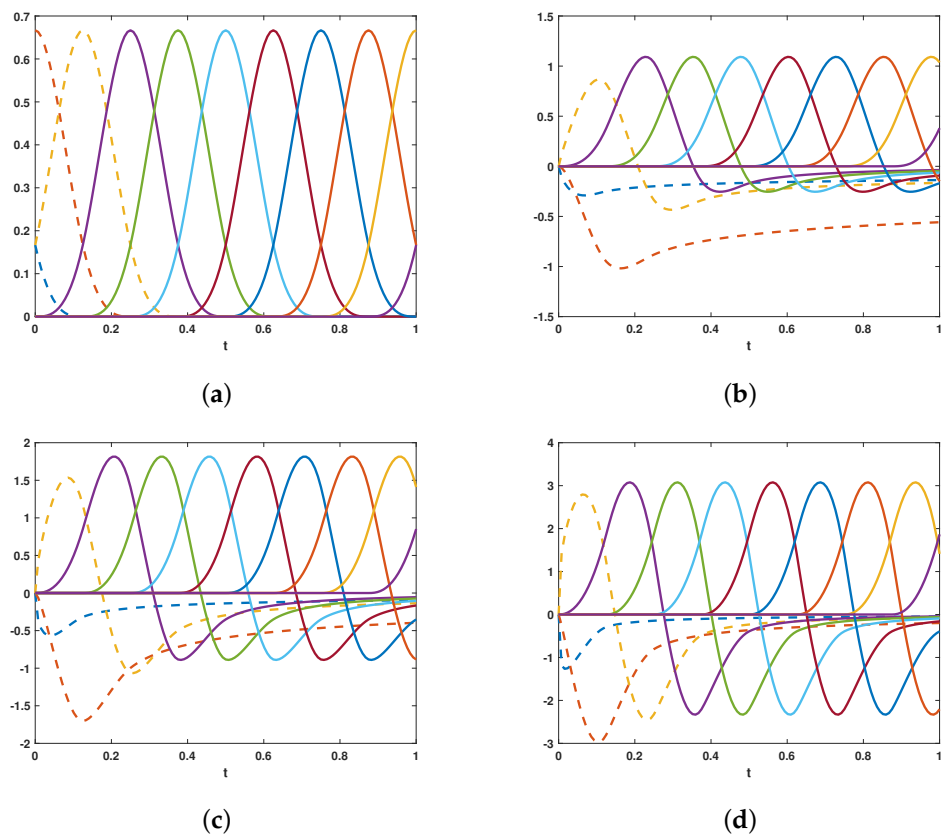
$${}_C D_t^\gamma B_{nj_k}(t) = 2^{\gamma j} \frac{\Delta^{n+1} T_{n-\gamma}(2^j t - k) - L_{nj_k}(2^j t)}{\Gamma(n + 1 - \gamma)}, \quad n \geq 1, \quad (33)$$

where:

$$L_{nj_k}(t) = \sum_{i=0}^{-k-1} (-1)^i \binom{n+1}{i} \left( (t-k-i)^{n-\gamma} + t^{1-\gamma} \sum_{p=0}^{n-1} \frac{(-1)^{n-p} (-k-i)^{n-1-p} (t-k-i)^p}{(n-1-p)!} \prod_{s=1}^{n-1-p} (\gamma-s) \right), \quad -n \leq k \leq -1. \quad (34)$$

We assume  $\prod_{s=1}^p (\gamma-s) = 1$  when  $p < 1$ .

In Figure 1, the cubic basis  $\mathcal{S}_3$  and its fractional derivatives at the refinement level  $j = 3$  are shown for some values of the fractional derivative  $\gamma$ .



**Figure 1.** (a) The cubic basis  $\mathcal{S}_3$  in the interval  $[0, 1]$  at the refinement level  $j = 3$ . The left edge functions are plotted with dashed lines. (b–d) The fractional derivative of the cubic basis for  $\gamma = 0.25$  (Panel (b)),  $\gamma = 0.5$  (Panel (c)), and  $\gamma = 0.75$  (Panel (d)).

#### 4. Numerical Results

In this section, we show the numerical solutions we obtain by the collocation method introduced in the previous section.

For the sake of simplicity, in the tests, we chose as collocation points the dyadic points of the interval  $I$ , i.e.,

$$\left\{ t_p = \frac{p}{2^s}, 0 \leq p \leq N_s \right\}, \quad (35)$$

where  $N_s = 2^s T$  and  $s \geq 0$  is the collocation level. Moreover, we used as approximating spaces the refinable spaces generated by the cubic B-spline basis  $\mathcal{S}_{3j}$ , and we set  $s = j + 1$  so that the time step is  $\Delta t = 2^{-j-1}$ . We used as approximating functions the two discrete quasi-interpolatory operators  $dQ_{3j}y$  obtained by setting  $n = 3$ ,  $r = 1$ , and  $r = 2$  in (23), i.e., the operators given in (26) and (27), respectively. The points  $\tau_{jk1}$  and  $\tau_{jk2}$  are the endpoints of  $\text{supp } B_{3jk}$ , i.e.,

$$\tau_{jk1} = \min(0, 2^{-j}k), \quad \tau_{jk2} = \max(2^{-j}(k+4), T), \quad -3 \leq k \leq N_j, \quad j \geq j_0.$$

To analyze the performance of the collocation method, we evaluate the infinity norm of the approximation error  $e_{3j}(t) = y(t) - y_{3j}(t)$ , i.e.,

$$\|e_{3j}\|_\infty = \max_{0 \leq p \leq 3N_s} |e_{3j}(t_p)|, \quad \left\{ t_p = \frac{p}{3 \cdot 2^s}, 0 \leq p \leq 3N_s \right\},$$

and the numerical convergence order defined as:

$$\rho_{3j} = \log_2 \left( \frac{\|e_{3,j-1}\|_\infty}{\|e_{3j}\|_\infty} \right).$$

For the ease of notation, in the following, we will drop the subscript 3.

##### 4.1. Example 1

In the first example, we set  $b = 0$ ,  $f(t) = \frac{\Gamma(\mu+1)}{\Gamma(\mu+1-\gamma)} t^{\mu-\gamma}$  and  $y(0) = 0$  so that the initial value problem (7) becomes:

$$\begin{cases} {}_C D_t^\gamma y(t) = \frac{\Gamma(\mu+1)}{\Gamma(\mu+1-\gamma)} t^{\mu-\gamma}, & 0 < t \leq 1, \quad 0 < \gamma < 1, \\ y(0) = 0. \end{cases}$$

In this case, the exact solution is  $y(t) = t^\mu$ . The discretization interval is  $I = [0, 1]$ .

Since the proposed method reproduces polynomials up to degree three when using the cubic B-spline basis, to check the polynomial reproducibility, in the first test, we set  $\mu = 3$ . Thus, the approximation is exact, and we expect the approximation error to be zero. Actually, the results listed in Table 1 show that the infinity norm of the error is in the order of machine precision for both quasi-interpolatory operators  $dQ_{3j}y$  with  $r = 1$  and  $r = 2$ .

To check the convergence order, in the second test, we set  $\mu = 2.8$  so that the theoretical convergence order is precisely  $\mu$ . The numerical convergence order for different values of  $\gamma$  is displayed in Figure 2 as a function of the refinement level  $j$ . For comparison, a line having the same slope as the theoretical convergence order is also displayed. The infinity norm of the approximation error and the numerical convergence order are listed in Tables 2 and 3. Table 2 refers to the quasi-interpolatory operators  $dQ_{3j}y$  with  $r = 1$ , while Table 3 refers to the case  $r = 2$ .

**Table 1.** Example 1: The infinity norm of the error  $\|e_j\|_\infty$  for  $\mu = 3$  and different values of the fractional derivative  $\gamma$  and the refinement level  $j$  when using  $dQ_{3j}y$  with  $r = 1$  and  $r = 2$  (cf. (26) and (27)). The number of collocation points is  $N_s + 1 = 2^j + 1$ .

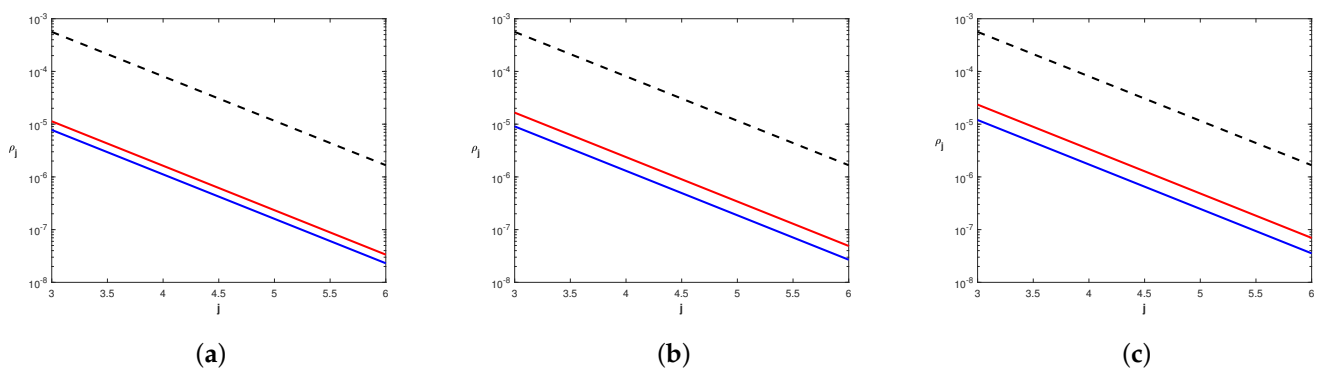
$j$	$N_s + 1$	$\gamma = 0.25$		$\gamma = 0.5$		$\gamma = 0.75$	
		$r = 1$	$r = 2$	$r = 1$	$r = 2$	$r = 1$	$r = 2$
3	17	$3.33 \times 10^{-16}$	$4.44 \times 10^{-16}$	$4.44 \times 10^{-16}$	$2.10 \times 10^{-16}$	$7.77 \times 10^{-16}$	$1.26 \times 10^{-15}$
4	33	$1.33 \times 10^{-15}$	$1.89 \times 10^{-15}$	$6.88 \times 10^{-15}$	$2.00 \times 10^{-15}$	$2.22 \times 10^{-15}$	$1.78 \times 10^{-15}$
5	65	$7.80 \times 10^{-14}$	$3.84 \times 10^{-14}$	$4.71 \times 10^{-14}$	$1.78 \times 10^{-14}$	$1.95 \times 10^{-14}$	$1.84 \times 10^{-14}$
6	129	$7.17 \times 10^{-13}$	$1.06 \times 10^{-12}$	$1.71 \times 10^{-13}$	$3.04 \times 10^{-13}$	$1.14 \times 10^{-13}$	$2.95 \times 10^{-13}$

**Table 2.** Example 1: The infinity norm of the error and the numerical convergence order for different values of the fractional derivative  $\gamma$  and the refinement level  $j$  when using  $dQ_{3j}y$  with  $r = 1$  (cf. (26)). The number of collocation points is  $N_s + 1 = 2^j + 1$ .

$j$	$N_s + 1$	$\gamma = 0.25$		$\gamma = 0.5$		$\gamma = 0.75$	
		$\ e_j\ _\infty$	$\rho_j$	$\ e_j\ _\infty$	$\rho_j$	$\ e_j\ _\infty$	$\rho_j$
3	17	$1.13 \times 10^{-5}$		$1.65 \times 10^{-5}$		$2.35 \times 10^{-5}$	
4	33	$1.63 \times 10^{-6}$	2.800	$2.38 \times 10^{-6}$	2.800	$3.37 \times 10^{-6}$	2.800
5	65	$2.34 \times 10^{-7}$	2.800	$3.41 \times 10^{-7}$	2.800	$4.84 \times 10^{-7}$	2.800
6	129	$3.35 \times 10^{-8}$	2.800	$4.90 \times 10^{-8}$	2.800	$6.96 \times 10^{-8}$	2.800

**Table 3.** Example 1: The infinity norm of the error and the numerical convergence order for different values of the fractional derivative  $\gamma$  and the refinement level  $j$  when using  $dQ_{3j}y$  with  $r = 2$  (cf. (27)). The number of collocation points is  $N_s + 1 = 2^j + 1$ .

$j$	$N_s + 1$	$\gamma = 0.25$		$\gamma = 0.5$		$\gamma = 0.75$	
		$\ e_j\ _\infty$	$\rho_j$	$\ e_j\ _\infty$	$\rho_j$	$\ e_j\ _\infty$	$\rho_j$
3	17	$7.77 \times 10^{-6}$		$9.07 \times 10^{-6}$		$1.20 \times 10^{-5}$	
4	33	$1.12 \times 10^{-6}$	2.800	$1.30 \times 10^{-6}$	2.800	$1.72 \times 10^{-6}$	2.800
5	65	$1.60 \times 10^{-7}$	2.800	$1.87 \times 10^{-7}$	2.800	$2.47 \times 10^{-7}$	2.800
6	129	$2.30 \times 10^{-8}$	2.800	$2.69 \times 10^{-8}$	2.800	$3.55 \times 10^{-8}$	2.800



**Figure 2.** Example 1: The numerical convergence order  $\rho_j$  for the quasi-interpolatory operator with  $r = 1$  (red solid line) and  $r = 2$  (blue solid line) for  $\gamma = 0.25$  (Panel (a)),  $\gamma = 0.5$  (Panel (b)), and  $\gamma = 0.75$  (Panel (c)) is displayed as a function of  $j$ . The black dashed line has the same slope as the theoretical convergence order.

#### 4.2. Example 2

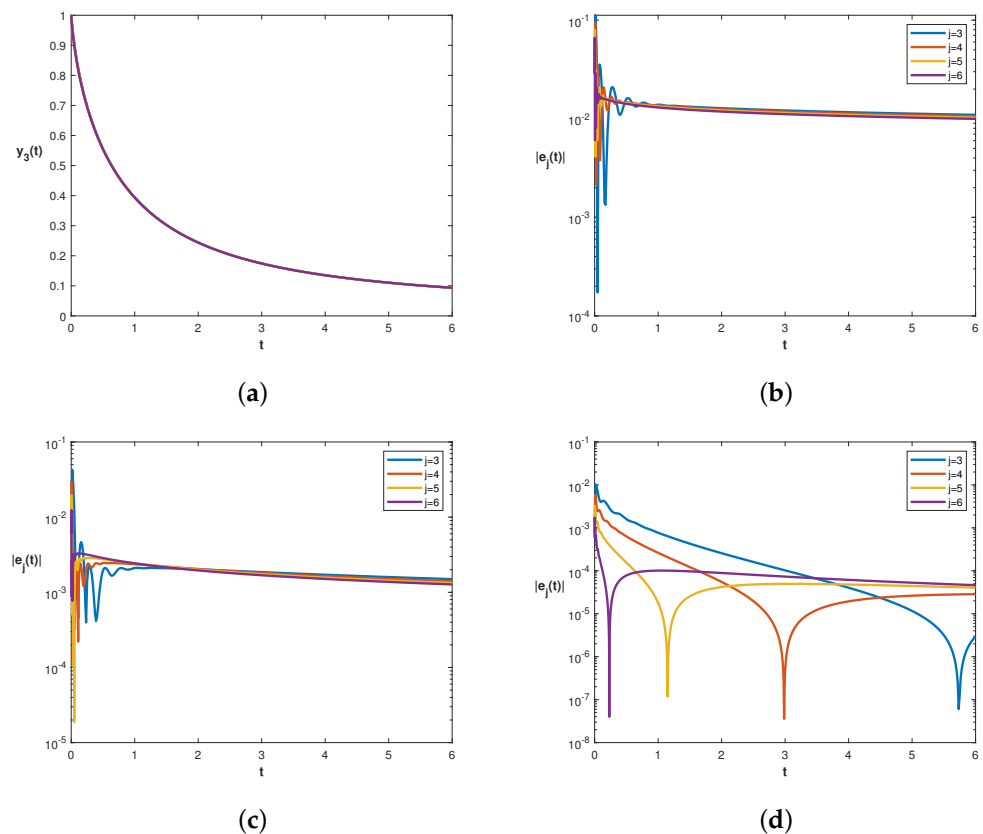
To check the accuracy of the method in the case of large intervals, in the second example, we set the discretization interval to  $I = [0, 6]$ . For this test, we chose  $f(t) \equiv 0$ ,  $b = 1$ , and  $y(0) = 1$ , so that the initial value problem (7) becomes:

$$\begin{cases} {}_C D_t^\gamma y(t) + y(t) = 0, & 0 < t \leq 6, \quad 0 < \gamma < 1, \\ y(0) = 1. \end{cases}$$

One can easily see that in this case, Equation (8) reduces to:

$$y(t) = E_{\gamma,1}(-t^\gamma). \quad (36)$$

To evaluate the Mittag–Leffler function appearing in the exact solution, we used the algorithm given in [37]. The numerical solution and the absolute value of the error for different values of  $j$  and  $\gamma$  are displayed in Figure 3. The numerical convergence order is shown in Figure 4. For comparison, a line having the same slope as the theoretical convergence order is also displayed. The infinity norm of the error and the numerical convergence order as a function of the refinement level  $j$  are listed in Tables 4 and 5. Table 4 refers to the quasi-interpolatory operators  $dQ_{3;j}y$  with  $r = 1$ , while Table 5 refers to the case  $r = 2$ .



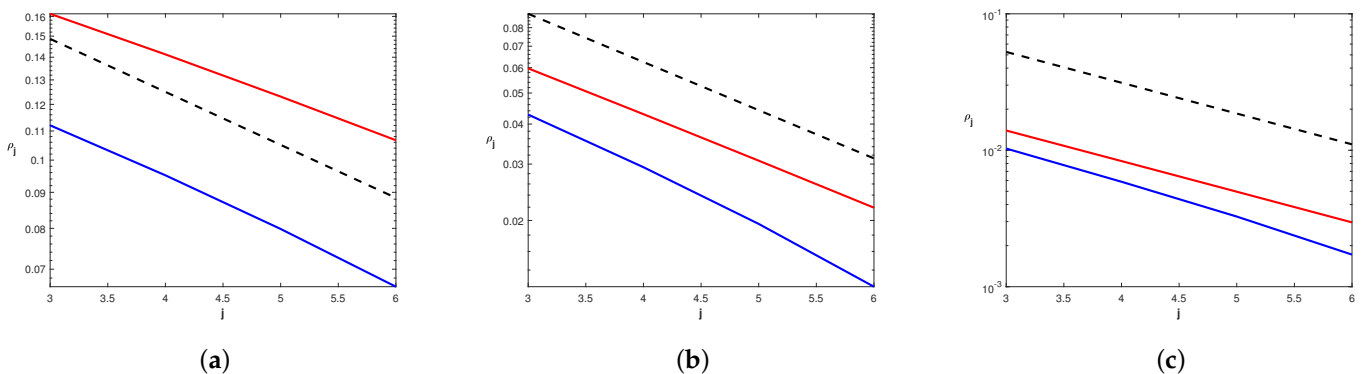
**Figure 3.** Example 2. (a) The numerical solution  $y_3(t)$  (blue solid line) and exact solution (red dashed line). (b–d) The semilog plot of the error  $|e_j(t)|$  for different values of  $j$  for  $\gamma = 0.25$  (Panel (b)),  $\gamma = 0.5$  (Panel (c)), and  $\gamma = 0.75$  (Panel (d)).

**Table 4.** Example 2: The infinity norm of the error and the numerical convergence order for different values of the fractional derivative  $\gamma$  and the refinement level  $j$  when using  $dQ_{3j}y$  with  $r = 1$  (cf. (26)). The number of collocation points is  $N_s + 1 = 2^j 6 + 1$ .

$j$	$N_s + 1$	$\gamma = 0.25$		$\gamma = 0.5$		$\gamma = 0.75$	
		$\ e_j\ _\infty$	$\rho_j$	$\ e_j\ _\infty$	$\rho_j$	$\ e_j\ _\infty$	$\rho_j$
3	97	$1.61 \times 10^{-1}$		$5.97 \times 10^{-2}$		$1.40 \times 10^{-2}$	
4	193	$1.41 \times 10^{-1}$	0.191	$4.29 \times 10^{-2}$	0.475	$8.33 \times 10^{-3}$	0.744
5	385	$1.23 \times 10^{-1}$	0.199	$3.07 \times 10^{-2}$	0.482	$4.97 \times 10^{-3}$	0.746
6	769	$1.07 \times 10^{-1}$	0.206	$2.19 \times 10^{-2}$	0.488	$2.96 \times 10^{-3}$	0.747

**Table 5.** Example 2: The infinity norm of the error and the numerical convergence order for different values of the fractional derivative  $\gamma$  and the refinement level  $j$  when using  $dQ_{3j}y$  with  $r = 2$  (cf. (27)). The number of collocation points is  $N_s + 1 = 2^j 6 + 1$ .

$j$	$N_s + 1$	$\gamma = 0.25$		$\gamma = 0.5$		$\gamma = 0.75$	
		$\ e_j\ _\infty$	$\rho_j$	$\ e_j\ _\infty$	$\rho_j$	$\ e_j\ _\infty$	$\rho_j$
3	97	$1.12 \times 10^{-1}$		$4.28 \times 10^{-2}$		$1.03 \times 10^{-2}$	
4	193	$9.52 \times 10^{-2}$	0.236	$2.93 \times 10^{-2}$	0.546	$5.88 \times 10^{-3}$	0.808
5	385	$7.98 \times 10^{-2}$	0.253	$1.95 \times 10^{-2}$	0.588	$3.27 \times 10^{-3}$	0.847
6	769	$6.61 \times 10^{-2}$	0.272	$1.24 \times 10^{-2}$	0.651	$1.72 \times 10^{-3}$	0.926



**Figure 4.** Example 2: The numerical convergence order  $\rho_j$  for the quasi-interpolatory operator with  $r = 1$  (red solid line) and  $r = 2$  (blue solid line) for  $\gamma = 0.25$  (Panel (a)),  $\gamma = 0.5$  (Panel (b)), and  $\gamma = 0.75$  (Panel (c)) is displayed as a function of  $j$ . The black dashed line has the same slope as the theoretical convergence order.

## 5. Discussion

The numerical results in Section 4 show that the collocation method we proposed is accurate and efficient. As shown in the first example of Section 4.1, the method is exact for polynomials up to degree  $n = 3$ , the degree of the spline approximating function, so that the error is in the order of the machine precision (cf. Table 1). In the second example of Section 4.1, the error decreases according to the theoretical convergence rate for both quasi-interpolatory operators. In fact, the numerical convergence rate is  $O(2^{-j\nu})$ , where  $\nu = 2.8$  is the smoothness of the solution  $y(t) = t^{2.8}$  (cf. Tables 2 and 3). Nevertheless, the quasi-interpolatory operator with  $r = 2$  gives a smaller error showing that higher accuracy can be achieved by operators with a higher degree of polynomial reproduction. It is worth noting that the error is rather small even at the low refinement level  $j = 3$ , i.e., using only 11 basis functions and 16 collocation points. A similar behavior is shown in the examples of Section 4.2 (cf. Tables 4 and 5). Here, we used a large discretization

interval, i.e.,  $[0, 6]$ . We point out that other numerical methods, such as spectral methods or finite difference methods, require high computational cost to approximate accurately the solution of fractional differential equations in large intervals. Instead, our method gives a good accuracy with a low computational cost. In this example, the numerical convergence rate is slightly different from the theoretical one, which is  $O(2^{-j\nu})$  with  $\nu = \gamma$ . In particular, the approximation with the quasi-interpolatory operator with  $r = 1$  shows a lower convergence rate while the quasi-interpolatory operator with  $r = 2$  shows a higher convergence rate. This behavior is worth investigating in more detail. Finally, we notice that the error is higher near the left boundary (cf. Figure 3). This can be due to the use of truncated B-splines as left-edge functions for which initial conditions are more difficult to fulfil. This problem can be overcome using optimal B-spline bases [38].

The proposed method can be easily applied to more general fractional differential equations. As an example, we consider the Bagley–Torvik equation:

$$\begin{cases} y''(t) + {}_c D_t^{1.5} y(t) + y(t) = \frac{15}{4} \sqrt{t} + \frac{15}{8} \sqrt{\pi} t + t^2 \sqrt{t}, & t \in [0, 1], \\ y(0) = y'(0) = 0, \end{cases} \quad (37)$$

which is a second order multi-term fractional differential equation used to describe the viscoelastic properties of real materials [39]. Its exact solution is  $y(t) = t^2 \sqrt{t}$ . We numerically solved Equation (37) using the quasi-interpolatory operator  $dQ_{nj}y$  with  $r = 1$  and  $r = 2$ . The infinity norm of the error and the numerical convergence order as a function of the refinement level  $j$  are listed in Table 6. We notice that in the case of differential problems having a second order derivative, the theoretical convergence order is smaller than  $O(2^{-j\nu})$ , as the numerical results show. The error is comparable with the error obtained by other spline methods in the literature (cf. [20,22]) and shows that the method is effective also for this kind of problem. More general problems, such as boundary value problems and nonlinear problems, will be the subject of a forthcoming paper.

**Table 6.** Bagley–Torvik equation: the infinity norm of the error and the numerical convergence order when using  $dQ_{nj}y$  with  $r = 1$  (left) and  $r = 2$  (right).

$r = 1$			$r = 2$		
$j$	$\ e_j\ _\infty$	$\rho_j$	$j$	$\ e_j\ _\infty$	$\rho_j$
3	$5.65 \times 10^{-2}$		3	$3.39 \times 10^{-3}$	
4	$2.89 \times 10^{-2}$	0.965	4	$1.15 \times 10^{-3}$	1.564
5	$1.48 \times 10^{-2}$	0.971	5	$3.87 \times 10^{-4}$	1.566
6	$7.50 \times 10^{-3}$	0.977	6	$1.30 \times 10^{-4}$	1.577

## 6. Conclusions

The numerical method presented in this paper combines collocation methods and spline quasi-interpolatory operators involving divided differences to approximate the solution to a linear time fractional differential equation. The method is convergent and reproduces polynomials of suitable degree. The numerical tests confirm the theoretical results and show that the accuracy of the method can be improved using quasi-interpolatory operators reproducing polynomials of higher degree.

The use of the proposed method to solve nonlinear fractional differential equations is at present under study. Moreover, we aim to improve the accuracy of the method by using optimal spline bases and different kinds of quasi-interpolatory operators.

**Author Contributions:** Conceptualization, F.P.; funding acquisition, F.P.; methodology, E.P., L.P., and F.P.; software, E.P.; writing, review and editing, E.P., L.P., and F.P. All authors read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by Gruppo Nazionale per il Calcolo Scientifico (Istituto Nazionale di Alta Matematica “Francesco Severi”), Grant INdAM–GNCS Project 2020 “Costruzione di metodi numerico/statistici basati su tecniche multiscale per il trattamento di segnali e immagini ad alta dimensionalità”.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data sharing not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Podlubny, I. *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*; Elsevier: Amsterdam, The Netherlands, 1998.
2. Sabatier, J.; Agrawal, O.; Tenreiro Machado, J.A. *Advances in Fractional Calculus*; Springer: Berlin/Heidelberg, Germany, 2007. [[CrossRef](#)]
3. Diethelm, K. *The Analysis of Fractional Differential Equations: An Application-Oriented Exposition Using Differential Operators of Caputo Type*; Springer: Berlin/Heidelberg, Germany, 2010.
4. Hilfer, R. *Applications of Fractional Calculus in Physics*; World Scientific: Singapore, 2000.
5. Kilbas, A.A.; Srivastava, H.M.; Trujillo, J.J. *Theory and Applications of Fractional Differential Equations*; Elsevier Science (North-Holland): Amsterdam, The Netherlands, 2006.
6. Mainardi, F. *Fractional Calculus and Waves in Linear Viscoelasticity: An Introduction to Mathematical Models*; World Scientific: Singapore, 2010.
7. Tarasov, V.E. *Fractional Dynamics*; Nonlinear Physical Science; Springer: Berlin/Heidelberg, Germany, 2010.
8. Baleanu, D.; Diethelm, K.; Scalas, E.; Trujillo, J.J. *Fractional Calculus: Models and Numerical Methods*; World Scientific: Singapore, 2016.
9. Caputo, M. Linear models of dissipation whose  $Q$  is almost frequency independent—II. *Geophys. J. Int.* **1967**, *13*, 529–539. [[CrossRef](#)]
10. Diethelm, K.; Garrappa, R.; Giusti, A.; Stynes, M. Why fractional derivatives with nonsingular kernels should not be used. *Fract. Calc. Appl. Anal.* **2020**, *23*, 610–634. [[CrossRef](#)]
11. Li, C.; Zeng, F. *Numerical Methods for Fractional Calculus*; CRC Press: Boca Raton, FL, USA, 2015; Volume 24.
12. Cai, M.; Li, C. Numerical approaches to fractional integrals and derivatives: A review. *Mathematics* **2020**, *8*, 43. [[CrossRef](#)]
13. Diethelm, K.; Ford, J.M.; Ford, N.J.; Weilbeer, M. Pitfalls in fast numerical solvers for fractional differential equations. *J. Comput. Appl. Math.* **2006**, *186*, 482–503. [[CrossRef](#)]
14. Li, C.; Chen, A. Numerical methods for fractional partial differential equations. *Int. J. Comput. Math.* **2018**, *95*, 1048–1099. [[CrossRef](#)]
15. Schumaker, L. *Spline Functions: Basic Theory*; Cambridge University Press: Cambridge, UK, 2007.
16. Unser, M.; Blu, T. Fractional splines and wavelets. *SIAM Rev.* **2000**, *42*, 43–67. [[CrossRef](#)]
17. Pezza, L.; Pitolli, F. A fractional spline collocation–Galerkin method for the time-fractional diffusion equation. *Commun. Appl. Ind. Math.* **2018**, *9*, 104–120. [[CrossRef](#)]
18. Pezza, L.; Pitolli, F. A multiscale collocation method for fractional differential problems. *Math. Comput. Simul.* **2018**, *147*, 210–219. [[CrossRef](#)]
19. Pitolli, F. A fractional B-spline collocation method for the numerical solution of fractional predator-prey models. *Fractal Fract.* **2018**, *2*, 13. [[CrossRef](#)]
20. Pedas, A.; Tamme, E. On the convergence of spline collocation methods for solving fractional differential equations. *J. Comput. Appl. Math.* **2011**, *235*, 3502–3514. [[CrossRef](#)]
21. Pedas, A.; Tamme, E. Numerical solution of nonlinear fractional differential equations by spline collocation methods. *J. Comput. Appl. Math.* **2014**, *255*, 216–230. [[CrossRef](#)]
22. Dadkhah, E.; Shiri, B.; Ghaffarzadeh, H.; Baleanu, D. Visco-elastic dampers in structural buildings and numerical solution with spline collocation methods. *J. Appl. Math. Comput.* **2020**, *63*, 29–57. [[CrossRef](#)]
23. Brunner, H. *Collocation Methods for Volterra Integral and Related Functional Differential Equations*; Cambridge University Press: Cambridge, UK, 2004; Volume 15.
24. Lyche, T.; Schumaker, L.L. Local spline approximation methods. *J. Approx. Theory* **1975**, *15*, 294–325. [[CrossRef](#)]
25. Gori, L.; Santi, E. Refinable quasi-interpolatory operators. In *Constructive Theory of Functions, Varna 2002*; Bojanov, B., Ed.; DARBA: Sofia, Bulgaria, 2003; pp. 288–294.

26. Sablonnière, P. Recent progress on univariate and multivariate polynomial and spline quasi-interpolants. In *Trends and Applications in Constructive Approximation, ISNM*; De Bruin, M., Mache, D., Szabados, J., Eds.; Birkhäuser Verlag: Basel, Switzerland, 2005; Volume 177, pp. 229–245.
27. Pellegrino, E.; Pezza, L.; Pitolli, F. A collocation method in spline spaces for the solution of linear fractional dynamical systems. *Math. Comput. Simul.* **2020**, *176*, 266–278. [[CrossRef](#)]
28. Pellegrino, E.; Pezza, L.; Pitolli, F. Quasi-interpolant operators and the solution of fractional differential problems. In *Approximation Theory XVI. Nashville 2019*; Fasshauer, G.E., Ed.; Springer Nature: Switzerland, 2020.
29. Gorenflo, R.; Kilbas, A.A.; Mainardi, F.; Rogosin, S.V. *Mittag–Leffler Functions, Related Topics and Applications*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 2.
30. de Boor, C. *A Practical Guide to Splines*; Springer: Berlin/Heidelberg, Germany, 1978.
31. Schoenberg, I.J. On spline functions. In *Inequalities*; Shisha, O., Ed.; Academic Press: New York, NY, USA, 1967; pp. 255–291.
32. de Boor, C.; Fix, G.J. Spline approximation by quasiinterpolants. *J. Approx. Theory* **1973**, *8*, 19–45. [[CrossRef](#)]
33. Goodman, T.N.T.; Sharma, A. A modified Bernstein–Schoenberg operator. In *Constructive Theory of Functions 87*; Sendov, B., Ed.; Bulgarian Academy Sciences: Sofia, Bulgaria, 1988.
34. Lee, B.G.; Lyche, T.; Mørken, K. Some examples of quasi-interpolants constructed from local spline projectors. In *Mathematical Methods for Curves and Surfaces. Oslo 2000*; Lyche, T., Schumaker, L., Eds.; Vanderbilt University Press: Nashville, TN, USA, 2001; pp. 243–252.
35. Gori, L.; Santi, E. Convergence properties of certain refinable quasi-interpolatory operators. *Appl. Numer. Math.* **2005**, *55*, 312–321. [[CrossRef](#)]
36. Ascher, U. Discrete least squares approximations for ordinary differential equations. *SIAM J. Numer. Anal.* **1978**, *15*, 478–496. [[CrossRef](#)]
37. Garrappa, R.; Popolizio, M. Evaluation of generalized Mittag–Leffler functions on the real line. *Adv. Comput. Math.* **2013**, *39*, 205–225. [[CrossRef](#)]
38. Pitolli, F. Optimal B-spline bases for the numerical solution of fractional differential problems. *Axioms* **2018**, *7*, 46. [[CrossRef](#)]
39. Torvik, P.J.; Bagley, R.L. On the appearance of the fractional derivative in the behaviour of real materials. *J. Appl. Mech.* **1984**, *51*, 294–298. [[CrossRef](#)]