THEORETICAL AND APPLIED MECHANICS
Article in Press (2026) DOI: https://doi.org/10.2298/TAM260219009M

TIME FRACTIONAL MODELS FOR SIGNAL SMOOTHING USING THE MULTIDOMAIN PSEUDOSPECTRAL METHOD

Nancy Mukwevho, Olumuyiwa Otegbeye, and Shina Daniel Oloniiju

Abstract. The processing and smoothing of noisy signals is a fundamental challenge in many scientific and engineering disciplines. Time-fractional diffusion models have recently emerged as powerful tools for addressing this challenge, offering significant advantages over classical smoothing methods by capturing memory effects and better preserving sharp features and edges. This study focuses on the numerical solution of such time-fractional models for one-dimensional signal smoothing. The primary source of computational difficulty in these models arises from the nonlocal nature of the temporal Caputo derivative of order γ, whose long-term memory kernel requires careful, accurate approximation over the entire time history. To address these challenges, this study develops a novel multidomain pseudospectral approach. Specifically, the temporal domain is discretized using the non-overlapping multidomain approach, while the spatial domain is handled using the overlapping domain decomposition strategy. The approximation of the function is composed of the Lagrange interpolating polynomials for the spatial domain and the shifted Chebyshev polynomial of the first kind for the temporal domain. The aim of this study is to combine these numerical components into a unified and robust scheme and to apply it, for the first time, to the problem of fractional-order signal smoothing. The study demonstrates the effectiveness of the developed pseudospectral multidomain method when applied to both linear and nonlinear time-fractional smoothing models, including the Rudin–Osher–Fatemi (ROF) diffusion model. The results are presented graphically and in a tabular form.

1. Introduction

The field of signal processing has seen remarkable advancements, playing a crucial role in a wide array of technologies, from medical imaging to wireless communications [1]. As technology progresses, signals with increasingly complex behaviors are being encountered, challenging the limitations of traditional signal processing

2020 *Mathematics Subject Classification:* Primary 65M70; Secondary 26A33, 94A12.
Key words and phrases: multidomain pseudospectral method, time-fractional smoothing model, overlapping spatial decomposition, non-overlapping temporal decomposition, Lagrange interpolating polynomials, Chebyshev polynomials.

techniques that rely on integer-order calculus [2]. These conventional methods often fall short in accurately representing the intricate dynamics of real-world signals, which frequently exhibit long-range correlations and memory effects [3]. To address these challenges, fractional calculus, a generalization of differentiation and integration to non-integer orders, has emerged as a powerful tool [4, 5, 6].

Fractional calculus provides a more flexible and accurate framework for modeling complex systems, particularly those with memory and hereditary properties [7, 8]. This has led to the development of fractional models, which have shown great promise in various signal processing applications. These models can effectively capture the non-local dependencies and self-similar patterns present in many signals. For instance, in image processing, fractional-order differential operators have demonstrated superiority over their integer-order counterparts in enhancing details such as weak edges and strong textures. Li et al. [9] used the time-fractional diffusion equation for signal smoothing. In a direct comparison, the time-fractional diffusion filtering approach not only surpassed classical diffusion filtering but, also outperformed established smoothing methods such as the regularization method, the Savitzky–Golay filter, and wavelet methods. The practical benefits were confirmed through its successful application in smoothing nuclear magnetic resonance (NMR) spectra, affirming its advantages for real-world signal processing challenges. Building on the success of the time-fractional diffusion equation (TFDE) for signal smoothing, the study by Jayaraj et al. [10] focused on optimizing its implementation. One significant challenge with the TFDE is its sensitivity to the time step size. To address this, a novel approach from Jayaraj et al. [10] introduced the hybrid Ebola-based reptile search (HERS) algorithm specifically to select the optimal time step, thereby enhancing the signal smoothing and peak-preservation capabilities of the diffusion filter. The optimized HERS-TFDE model was also validated against the wavelet model, Savitzky-Golay, and regularization techniques, and its superiority was confirmed.

The utility of fractional concepts is remarkably diverse. Beyond the differential models, some methodologies use fractional-octave profiles, for example, Hatziantoniou and Mourjopoulos [11] introduced a methodology based on fractional-octave profiles. The approach introduced a novel time-frequency mapping approach to smooth the complex transfer function of measured responses, deriving a corresponding impulse response that is both less complex and consistent with perceptual principles. However, the practical success of fractional differential operators is especially evident in filtering and enhancement tasks across various domains. The success of fractional operators is especially prominent in digital image processing, a domain where traditional integer-order partial differential equation (PDE) models often fail by blurring important edge and texture details. To overcome this, fractional partial differential equation models were developed by Pu et al. [12], using the unique ability of fractional operators to nonlinearly enhance complex textures while removing noise. This has led to significant theoretical work, including the

development of fractional Green's and Euler–Lagrange formulas specifically for two-dimensional image processing. The presented results proved that the proposed denoising model preserves the high-frequency edge and complex texture information than the traditional integral-based algorithms.

In digital image processing, for instance, an improved model based on the Grünwald-Letnikov (G-L) fractional differential operator was proposed by He et al. [13]. The resulting nonlinear filter masks were capable of enhancing and preserving detailed features like textures while effectively removing noise. The methods have demonstrated the capabilities not only on texture-rich digital images, but have also been successfully used to denoise three-dimensional magnetic resonance images (MRI). Similarly, time-space fractional-order models, which are a generation of the classical diffusion model, have proven to be excellent for signal smoothing and peak-preserving. The time-fractional derivative can model particle sticking and trapping phenomena, while the space-fractional derivative can describe long particle jumps, making these models suitable for describing anomalous diffusion processes [14]. Recent research has highlighted the practical success of these models. For example, in the effort to enhance signal quality by filtering out noise in transient electromagnetic (TEM) signals, Tan et al. [15] proposed a time-space fractional-order diffusion model (TSFDM) that uses intelligent optimization. The approach used a discrete difference method for the time Caputo derivative and a Grünwald-Letnikov approximation for the space derivative to establish an iterative difference equation. A key innovation was the use of an enhanced Harris Hawk optimization algorithm to find the optimal parameters for the TSFDM filter. The results were compelling, with simulation experiments demonstrating a significant signal-to-noise ratio increase of $33dB$, showcasing superior performance compared to methods such as empirical mode decomposition (EMD) and variational mode decomposition (VMD), validating the model's effectiveness in a real-world application. Wang et al. [16] proposed a novel method that uses the distinct properties of both fractional differentiation and integration, first using a fractional differential gradient to precisely detect the location of noise points, and then applying fractional integration to denoise the image. The dual approach was effective, as fractional integration can nonlinearly suppress high-frequency noise while retaining the low-frequency information corresponding to the image's essential features, thus maintaining critical edge and texture details. Huang et al. [17] examined the image enhancement and denoising using fractional order models, focusing on the effectiveness of fractional differential operators in enhancing image features and reducing noise. The presented results confirmed the effectiveness of the fractional-order image processing model in enhancing contrast and clarity against integer-order models. This concept of feature enhancement is critical in high-stakes applications like medical diagnosis. Wu et al. [18] used a two-dimensional fractional-order convolution (FOC) as a key component in a multilayer machine vision classifier designed to automatically identify lung diseases from chest X-ray images. The FOC approach enhanced the symptomatic textural features of diseases while simultaneously suppressing noise, providing a cleaner input for the subsequent classification stages. The high accuracy and recall

of the complete system underscored the value of fractional operators as a powerful pre-processing step in automated medical diagnostics.

Despite their modeling advantages, fractional differential equations are often analytically intractable, necessitating the development of robust and specialized numerical methods. The research landscape for these methods is rich and varied. While approaches such as the finite difference method used by Tan et al. [15] are common, other sophisticated techniques are being explored. Ma et al. [19] focused on systems of fuzzy fractional differential equations with a lateral type of the Hukuhara derivative and the generalized Hukuhara derivative to model signal smoothing, using a collocation method on discontinuous piecewise polynomial spaces.

Among various numerical techniques, spectral methods, particularly the pseudospectral method, have gained prominence due to their high accuracy and efficiency, especially for problems with smooth solutions [20]. These methods are well-suited for handling the non-local nature of fractional differential operators [21]. The pseudospectral method leverages global basis functions, such as Chebyshev or Legendre polynomials, to approximate the solution, which can lead to exponential convergence rates for smooth problems [20, 21].

However, for problems involving long-time integration or signals with varying characteristics across the domain, single-domain pseudospectral methods can become less efficient. To overcome this limitation, the multidomain pseudospectral method has been introduced. This domain decomposition allows for greater flexibility in adapting to the local behavior of the signal and can lead to more accurate and efficient solutions, particularly for problems with long-term dynamics [21].

This study focuses on the development and application of a multidomain pseudospectral method for solving time-fractional models in the context of signal smoothing. By combining the descriptive power of fractional calculus with the numerical efficiency of the multidomain pseudospectral method, this study aims to provide a robust and accurate approach for analyzing and processing complex signals. The proposed technique discretizes the spatial dimension using Lagrange interpolating polynomials and the temporal dimension using shifted Chebyshev polynomials of the first kind. The spatial domain is divided into overlapping equal subdomains, and the temporal domain is divided into non-overlapping equal subdomains. The proposed scheme is designed to overcome the challenges associated with solving fractional-order partial differential equations and to offer a superior alternative to traditional signal smoothing techniques.

2. Preliminaries

This section introduces the core components of fractional calculus and the Chebyshev pseudospectral method, which are fundamental building blocks for the scheme developed in the subsequent section.

Definition 2.1. Suppose $u\colon ([a,b]\times[0,h]) \longrightarrow \mathbb{R}$ is a continuous function in time for each fixed $x \in [a,b]$, and let $\gamma > 0$. The *Riemann–Liouville fractional integral* of order γ in time is defined as [21]

$$ {}_0^{RL}I_t^{\gamma}u(x,t) = \begin{cases} \frac{1}{\Gamma(\gamma)}(t^{\gamma-1} * u(x,t)) = \frac{1}{\Gamma(\gamma)}\int_0^t (t-\tau)^{\gamma-1}u(x,\tau)d\tau, & \gamma > 0, \\ u(x,t), & \gamma = 0, \end{cases} $$

where Γ is the Euler gamma function and $\big(t^{\gamma-1} * u(x,t)\big)$ denotes the convolution product of the power function $t^{\gamma-1}$ and $u(x,t)$.

Definition 2.2. The Caputo fractional differential operator of order $\gamma > 0$ applied to the function $u(x,t)$, where $u(x,t) \in C_t^{\lceil\gamma\rceil}([a,b]\times[0,h])$, the space of $\lceil\gamma\rceil$-times continuously differentiable functions in t on $[0,h]$, for each $x \in [a,b]$, is defined as [21]

$$ {}_0^{C}D_t^{\gamma}u(x,t) = {}_0^{RL}I_t^{\lceil\gamma\rceil-\gamma}\Big(\frac{\partial^{\lceil\gamma\rceil}u(x,t)}{\partial t^{\lceil\gamma\rceil}}\Big) = \frac{1}{\Gamma(\lceil\gamma\rceil-\gamma)}\int_0^t (t-\tau)^{\lceil\gamma\rceil-\gamma-1}\frac{\partial^{\lceil\gamma\rceil}u(x,\tau)}{\partial\tau^{\lceil\gamma\rceil}}d\tau, $$

for $\lceil\gamma\rceil - 1 < \gamma \leqslant \lceil\gamma\rceil$.

Definition 2.3. The discretization of the spatial $[a,b]$ and temporal domains $[0,h]$ relies on Chebyshev-Gauss-Lobatto (CGL) points. The standard CGL points, originally defined on the interval $[-1,1]$, are mapped to the physical domains using the affine transformations defined as [22]:

$$ x_e = \frac{b-a}{2}\cos\Big(\frac{\pi e}{N}\Big) + \frac{a+b}{2}, \quad 0 \leqslant e \leqslant N, $$
$$ t_r = \frac{h}{2}\Big(1+\cos\Big(\frac{\pi r}{M}\Big)\Big), \quad 0 \leqslant r \leqslant M. $$

Definition 2.4. Shifted Chebyshev polynomials defined on $t \in [0,h]$ are defined by the recurrence relation

$$ \Pi_{h,0}(t) = 1, \quad \Pi_{h,1}(t) = \frac{2t}{h} - 1, \quad \Pi_{h,j+1}(t) = 2\Pi_{h,1}(t)\cdot\Pi_{h,j}(t) - \Pi_{h,j-1}(t), \quad j \geqslant 1, $$

and the polynomials can be expanded in series form as [23]

$$ \Pi_{h,j}(t) = j\sum_{k=0}^{j}\frac{(-1)^{j-k}(j+k-1)!2^{2k}}{(j-k)!(2k)!h^k}t^k. $$

The Shifted Chebyshev polynomials satisfy the orthogonality condition

$$ \langle \Pi_{h,j}(t), \Pi_{h,m}(t)\rangle = \frac{1}{h_j}\int_0^h \Pi_{h,m}(t)\Pi_{h,j}(t)w_h(t)dt = \delta_{mj}, $$

where the weight function is given by $w(t) = \frac{1}{\sqrt{ht-t^2}}$.

Definition 2.5. In the pseudospectral method, a function $u(x)$ is approximated by a unique N^{th} degree polynomial, $U_N(x)$, that interpolates the function at the $N+1$ CGL nodes $\{x_e\}_{e=0}^N$. This interpolant can be expressed using the Lagrange cardinal polynomials, $\mathcal{L}(x)$, as

$$ u(x) \approx U_N(x) = \sum_{n=0}^{N} U(x_n)\mathcal{L}_n(x_e). $$

The Lagrange cardinal polynomials are defined by the

$$\mathcal{L}_n(x) = \prod_{\substack{e=0 \\ e\neq n}}^{N} \frac{x - x_e}{x_n - x_e}, \quad \text{such that} \quad \mathcal{L}_n(x_e) = \delta_{en} = \begin{cases} 1, & e = n, \\ 0, & e \neq n. \end{cases}$$

3. Numerical discretization of time-space fractional differential equations

This section develops a multidomain pseudospectral method for solving time-fractional diffusion equations arising in signal smoothing. The proposed approach combines two complementary decomposition approaches:

- Overlapping decomposition of the spatial domain, where the differential operator is classical, and
- Non-overlapping decomposition of the temporal domain, where the temporal evolution is governed by a Caputo fractional derivative.

This setting captures realistic signal-processing scenarios, where spatial variability is often localized while temporal dynamics exhibit long-range memory.

THEOREM 3.1 (Spectral Collocation Approximation). *Let $u(x,t)$ be a sufficiently smooth function defined on $([a,b]\times[0,h])$. The function $u(x,t)$ can be approximated at an arbitrary point (x_e,t_r) as*

$$U_{M,N}(x_e,t_r) = [\mathbf{E}\otimes\mathbf{I}_x]\mathbf{U}^{\dagger}_{M,N},$$

where $\mathbf{E} = \boldsymbol{\rho}\,\boldsymbol{\Pi}_h^{\mathsf{T}}\,\boldsymbol{\vartheta}\,\boldsymbol{\Pi}_h$ and $\mathbf{I}_x$ is of size $(N+1)\times(N+1)$.

PROOF. Since $u(x,t)$ is sufficiently smooth, it can be expanded in terms of shifted Chebyshev polynomials $\Pi_{h,j}(t)$ and the Lagrange polynomials $\mathcal{L}_n(x)$ and collocated at $\{(x_e,t_r)\in([a,b]\times[0,h])\}$, as

$$u(x,t) \approx U_{M,N}(x_e,t_r) = \sum_{n=0}^{N}\sum_{j=0}^{M}\frac{1}{\nu_j}\sum_{i=0}^{M} U_{M,N}(x_n,t_i)\,\Pi_{h,j}(t_i)\,\varrho_i\,\Pi_{h,j}(t_r)\,\mathcal{L}_n(x_e). \tag{3.1}$$

We define

$$E_i(t_r) = \varrho_i \sum_{j=0}^{M}\frac{1}{\nu_j}\Pi_{h,j}(t_i)\Pi_{h,j}(t_r),$$

then Equation (3.1) becomes

$$U_{M,N}(x_e,t_r) = \sum_{n=0}^{N}\sum_{i=0}^{M} U_{M,N}(x_n,t_i)E_i(t_r)\mathcal{L}_n(x_e), \quad 0\leqslant r\leqslant M, \quad 0\leqslant e\leqslant N,$$

which can be rewritten in matrix-vector form as

$$U_{M,N}(x_e,t_r) = [\mathbf{E}\otimes\mathbf{I}_x]\,\mathbf{U}^{\dagger}_{M,N}.$$

where $\mathbf{E}$ is the temporal interpolation matrix defined in the statement of Theorem 3.1. For completeness, its constituent matrices are:

$$\boldsymbol{\rho} = \begin{bmatrix} \varrho_0 & & & \\ & \varrho_1 & & \\ & & \ddots & \\ & & & \varrho_M \end{bmatrix}, \quad \boldsymbol{\vartheta} = \begin{bmatrix} \frac{1}{\nu_0} & & & \\ & \frac{1}{\nu_1} & & \\ & & \ddots & \\ & & & \frac{1}{\nu_M} \end{bmatrix},$$

$$\boldsymbol{\Pi}_h = \begin{bmatrix} \Pi_{h,0}(t_0) & \Pi_{h,0}(t_1) & \dots & \Pi_{h,0}(t_M) \\ \Pi_{h,1}(t_0) & \Pi_{h,1}(t_1) & \dots & \Pi_{h,1}(t_M) \\ \vdots & \vdots & \ddots & \vdots \\ \Pi_{h,M}(t_0) & \Pi_{h,M}(t_1) & \cdots & \Pi_{h,M}(t_M) \end{bmatrix}.$$

Here, $\mathbf{U}^{\dagger}_{M,N} \in \mathbb{R}^{MN+M+N+1}$ is the row-wise vectorization of the matrix $\mathbf{U}_{M,N} \in \mathbb{R}^{(M+1)\times(N+1)}$ defined as

$$\mathbf{U}_{M,N} = \begin{bmatrix} U_{M,N}(x_0,t_0) & U_{M,N}(x_1,t_0) & \dots & U_{M,N}(x_N,t_0) \\ U_{M,N}(x_0,t_1) & U_{M,N}(x_1,t_1) & \dots & U_{M,N}(x_N,t_1) \\ \vdots & \vdots & \vdots & \vdots \\ U_{M,N}(x_0,t_M) & U_{M,N}(x_1,t_M) & \dots & U_{M,N}(x_N,t_M) \end{bmatrix}.$$

This completes the proof. □

3.1. Partitioning of the spatial domain. The spatial domain $x \in [a,b]$ is divided into p overlapping equal-length subdomains. For mathematical illustration, each subdomain Ω_l is defined as follows:

$$\Omega_l = \left[x^{(l-1)}, \tilde{x}^{(l)}\right], \quad x^{(l-1)} < x^{(l)} < \tilde{x}^{(l)}, \quad x_0^{(0)} = a, \ \tilde{x}^{(q)} = b, \ \ l = 1,2,3,\dots,p.$$

The domain $\bar{x} \in [x^{(l-1)}, \tilde{x}^{(l)}]$ of each l-th subinterval is mapped onto $x \in [0, \mathbf{X}]$ using the following linear transformation:

$$x(\bar{x}) = \frac{\mathbf{X}(\bar{x} - x^{(l-1)})}{\tilde{x}^{(l)} - x^{(l-1)}} = \frac{\mathbf{X}(\bar{x} - x^{(l-1)})}{\mathbf{X}} = \bar{x} - x^{(l-1)}, \quad \bar{x} \in [x^{(l-1)}, \tilde{x}^{(l)}].$$

Note that the length of each subdomain $[x^{(l-1)}, \tilde{x}^{(l)}]$ is $\mathbf{X}$, i.e. $\mathbf{X} = \tilde{x}^{(l)} - x^{(l-1)}$. The intersection (overlap) between the subintervals occurs at the penultimate collocation point of the previous subinterval and the first collocation point of the current subinterval. That is, the penultimate collocation point in subdomain, Ω_{l-1}, will coincide with the first grid point of subdomain, Ω_l. See Figure 7 in Appendix A for a pictorial representation of the decomposition of the domain $[a,b]$ into p overlapping subdomains. Also, note that p overlapping subintervals result in $p-1$ overlapping regions.

3.2. Partitioning of the temporal domain. The temporal domain $t \in [0,h]$, on the other hand, is decomposed into q equal non-overlapping subintervals as follows:

$$h_l = [t_{l-1}, t_l], \quad 0 = t_0 < t_1 < \ldots < t_q = h, \quad l = 1,\dots,q.$$

Each subinterval is then discretized using $M+1$ distinct CGL points, $\{0 = t_0^{(1)}, \ldots, t_M^{(1)} = t_0^{(2)}, \ldots, t_M^{(l-1)} = t_0^{(l)}, \ldots, t_M^{(q)} = h\}$. Figure 8 in Appendix C illustrates the partitioning of the temporal domain $[0, h]$ into q equal, non-overlapping subintervals.

3.3. The Complete Discretization Grid. The partitioning for both the spatial and temporal domains separately can be combined to form a complete picture of the two-dimensional computational grid where the solution $U(x,t)$ is approximated. The spatial domain $[a,b]$ is represented along the horizontal axis, while the temporal domain $[0,h]$ is represented along the vertical axis. The set of all spatial collocation points defines a series of vertical grid lines, and the set of all temporal collocation points defines a series of horizontal grid lines. The intersections of these lines, (x_e, t_r), form the discrete grid where the numerical solution is computed.

Figure 9 in Appendix D illustrates the complete structure of the two-dimensional computational grid. The overlapping vertical bands correspond to the spatial subdomains Ω_l as shown in Figure 7, while the non-overlapping horizontal bands correspond to the temporal subdomains h_l from Figure 8. The intersection of a spatial and a temporal band forms a local computational block, which partitions the entire solution domain $[a,b] \times [0,h]$, and the solution $U(x,t)$ is computed at the intersections of the grid lines.

THEOREM 3.2 (Global spatial differential matrix on an overlapping grid). *Let $u(x,t)$ be a sufficiently smooth function defined on $[a,b]$, and let the domain be partitioned as per Section 3.1. Let the function $u(x,t)$ within each subdomain Ω_l be approximated by a unique Lagrange interpolating polynomial as defined in Definition 2.5 of degree N based on the CGL points within that subdomain. Then, the first derivative of $u(x,t)$ at the P unique grid points across the entire domain can be approximated at the collocation nodes x_e and t_r as*

$$\frac{\partial U_{M,N}(x_e, t_r)}{\partial x} \approx [\mathbf{E} \otimes \mathbf{D}]\mathbf{U}^{\dagger}_{\mathbf{M},\mathbf{N}},$$

where $\mathbf{D}$ is the $P \times P$ global spatial differentiation matrix.

PROOF. Generally, the collocation points across the entire x-domain can be represented as a set that spans all points within each subdivided region. By dividing the domain into multiple intervals, we establish a structured set of points that comprehensively covers the entire x-domain as follows

$$\begin{aligned}\Big\{a = x_0^{(0)}, x_1^{(0)}, \ldots, x_N^{(0)} = x_1^{(1)}, x_2^{(1)}, \ldots, x_N^{(1)} = x_1^{(2)}, x_2^{(2)}, \ldots, \\ x_N^{(2)} = x_1^{(3)}, x_2^{(3)}, \ldots, x_N^{(l-1)} = x_1^{(l)}, x_2^{(l)}, \ldots, x_N^{(l)} = x_1^{(l+1)}, \\ x_2^{(l+1)}, \ldots, x_N^{(p-2)} = x_1^{(p-1)}, x_2^{(p-1)}, \ldots, x_N^{(p-1)} = b\Big\}, \qquad 4 \leqslant l \leqslant p-3.\end{aligned}$$

where $x_0^{(0)} = a$ and $x_N^{(p-1)} = b$, define the leftmost and rightmost boundaries of the overall domain. For each subinterval $l = 1, 2, \ldots, p$, the grid points $x_0^{(l)}, x_1^{(l)}, \ldots, x_N^{(l)}$ proceed from left to right, ensuring continuity between adjacent subintervals. Specifically, $x_N^{(l-1)} = x_1^{(l)}$ for $l = 1, 2, \ldots, p-1$. This arrangement maintains

the left-to-right ordering of collocation points within each subinterval, Ω_l in the x-domain. Since there are p subdomains and $N+1$ collocation points in each, but adjacent subdomains share two points at the interface. That is, the first and last subdomains retain N unique collocation points each, while the remaining $p-2$ subdomains contribute $N-1$ distinct collocation points due to shared boundaries. Therefore, this brings the total number of distinct collocation points across the entire domain to $P = p(N-1)+2$.

It remains to find an explicit formula for the length of each subdomain in terms of the number of subintervals, p. If the entire domain of the differential equation is $[a,b]$, then the total length, $(b-a)$, of the domain is defined as

$$\begin{aligned}(b-a) = &(\text{number of subintervals} \times \text{length of each subinterval})\\ &- (\text{number of overlapping regions} \times \text{length of overlapping region}).\end{aligned}$$

where the length of the overlapping regions is given as

$$\begin{aligned}x_N^{(l)} - x_{N-1}^{(l)} = \frac{\mathbf{X}}{2}\Big(1-\cos\Big(\frac{\pi N}{N}\Big)\Big) - \frac{\mathbf{X}}{2}\Big(1-\cos\Big(\frac{\pi(N-1)}{N}\Big)\Big)\\ = \frac{\mathbf{X}}{2}\Big(1+\cos\Big(\frac{\pi(N-1)}{N}\Big)\Big).\end{aligned}$$

Therefore, the total length of the entire domain $(b-a)$ is now obtained as

$$(b-a) = p\mathbf{X} - (p-1)\frac{\mathbf{X}}{2}\Big(1+\cos\Big(\frac{\pi(N-1)}{N}\Big)\Big),$$

which implies that the length, $\mathbf{X}$, of each subdomain is now defined as

$$\mathbf{X} = \frac{2(b-a)}{2p-(p-1)\big(1+\cos\big(\frac{\pi(N-1)}{N}\big)\big)}.$$

Since the solution is collocated at $N+1$ spatial points, we apply the derivative at (x_e, t_r) by constructing the spectral differentiation matrix $\mathbf{D}$, on interval $[a,b]$, defined as

$$\mathbf{D} = \frac{2}{b-a}\hat{\mathbf{D}}.$$

The matrix $\hat{\mathbf{D}} \in \mathbb{R}^{(N+1)\times(N+1)}$ is the standard differentiation matrix on the interval $[-1,1]$, whose entries are given in [24]. Therefore, at the collocation points, the first-order derivative of $u(x,t)$ with respect to x is approximated as follows

$$\frac{\partial U_{M,N}(x_e,t_r)}{\partial x} = \sum_{n=0}^{N}\sum_{i=0}^{M} U_{M,N}(x_n,t_i)E_i(t_r)D_{e,n}, \tag{3.2}$$

where $D_{e,n}$ are entries of $\mathbf{D}$, and Equation (3.2), in matrix-vector form, becomes

$$\frac{\partial U_{M,N}(x_e,t_r)}{\partial x} = [\mathbf{E}\otimes\mathbf{D}]\mathbf{U}_{M,N}^{\dagger},$$

where $\mathbf{E}$ and $\mathbf{U}_{M,N}$ are defined as in Theorem 3.1. Given that the partitioning of the spatial domain $[a,b]$ into p overlapping subdomains, then the total number of distinct spatial collocation points is $P = p(N-1)+2$, and the differentiation of $u(x.t)$ with respect to x, the matrix $\mathbf{D}$ representing the entire spatial domain is an

$P \times P$ matrix defined in Appendix B. The matrix of the solution at all unique grid points across the entire domain is of size $(M+1) \times P$ defined as

$$\tilde{\mathbf{U}}_{M,P} = \begin{bmatrix} U(\hat{x}_1, t_0) & U(\hat{x}_2, t_0) & \dots & U(\hat{x}_P, t_0) \\ U(\hat{x}_1, t_1) & U(\hat{x}_2, t_1) & \dots & U(\hat{x}_P, t_1) \\ \vdots & \vdots & \ddots & \vdots \\ U(\hat{x}_1, t_M) & U(\hat{x}_2, t_M) & \dots & U(\hat{x}_P, t_M) \end{bmatrix},$$

where $\{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_P\}$ represents the ordered set of all P distinct collocation points from a to b, with $\hat{x}_1 = x_0$ and $\hat{x}_P = x_N$ and $\tilde{\mathbf{U}}^{\dagger}_{M,P}$ is the row-wise vectorization. Then, the matrix-vector representation $U(x_e, t_r)$

$$U(x_e, t_r) = [\mathbf{E} \otimes \mathbf{D}]\tilde{\mathbf{U}}^{\dagger}_{M,P}.$$

The higher-order spatial derivatives of $u(x,t)$ can be approximated by taking powers of the spectral differentiation matrix, $\mathbf{D}$. Generally, the w-th order spatial derivative, for any integer $w \geqslant 1$, is approximated as

$$\frac{\partial^w U_{M,N}(x_e, t_r)}{\partial x^w} = [\mathbf{E} \otimes \mathbf{D}^w]\, \mathbf{U}^{\dagger}_{M,N}.$$

This completes the proof. □

THEOREM 3.3 (Global temporal fractional derivative operator). *For a function defined on the complete space-time grid, the temporal Caputo fractional derivative ${}_0^C D_t^{\gamma} u(x,t)$ within the l-th time subdomain is approximated as*

$$(3.3)\quad {}_0^C D_t^{\gamma} u(x,t) \approx \sum_{e=1}^{l-1} \left[[({}_{t_{e-1},t_e}\mathbf{D}^{\gamma})^{\mathsf{T}} \otimes \mathbf{I}_P] \left(\tilde{\mathbf{U}}^{(e)}_{M,P}\right)^{\dagger} \right] + [({}_{t_{l-1},t_l}\mathbf{D}^{\gamma})^{\mathsf{T}} \otimes \mathbf{I}_P] \left(\tilde{\mathbf{U}}^{(l)}_{M,P}\right)^{\dagger},$$

where $\tilde{\mathbf{U}}^{(l)}_{M,P}$ encodes the full space-time solution on the l-th subdomain across all P spatial grid points and $\mathbf{I}_P \in \mathbb{R}^{P \times P}$ denotes the identity matrix associated with the P global spatial collocation points. In the first subinterval ($l = 1$), the memory term is naturally absent, and only the local contribution is used to numerically approximate the fractional derivative of $u(x,t)$.

PROOF. Let $u(x,t)$ be a sufficiently smooth function defined on the domain $([a,b] \times [0,h])$, and let $\gamma \in \mathbb{R}^+$ such that $\lceil \gamma \rceil - 1 < \gamma \leqslant \lceil \gamma \rceil$. If the time domain $[0,h]$ is partitioned into q equal, non-overlapping subintervals $h_l = [t_{l-1}, t_l]$, where $0 = t_0 < t_1 < \cdots < t_q = h$, then on each subinterval h_l, we approximate the Caputo fractional derivative of $u(x,t)$. That is, for $t \in [t_{l-1}, t_l]$, the Caputo fractional derivative, ${}_0^C D_t^{\gamma} u(x,t)$, is represented as follows

$$(3.4)\quad {}_0^C D_t^{\gamma} u(x,t) = \sum_{e=1}^{l-1} \left[\frac{1}{\Gamma(\lceil \gamma \rceil - \gamma)} \int_{t_{e-1}}^{t_e} (t-\tau)^{\lceil \gamma \rceil - \gamma - 1} \frac{\partial u^{\lceil \gamma \rceil}(x,\tau)}{\partial \tau^{\lceil \gamma \rceil}} d\tau \right] + \frac{1}{\Gamma(\lceil \gamma \rceil - \gamma)} \int_{t_{l-1}}^{t} (t-\tau)^{\lceil \gamma \rceil - \gamma - 1} \frac{\partial u^{\lceil \gamma \rceil}(x,\tau)}{\partial \tau^{\lceil \gamma \rceil}} d\tau.$$

The Caputo fractional derivative is split into two components: a *local term*, which is the second term of Equation (3.4), and a *memory term*, the first term of Equation

(3.4). This decomposition holds across the entire partitioned spatial domain. The local term is approximated over the current time subinterval h_l for all S spatial points, and its discrete form is given as

$$\frac{1}{\Gamma(\lceil\gamma\rceil-\gamma)}\int_{t_{l-1}}^{t}(t-\tau)^{\lceil\gamma\rceil-\gamma-1}\frac{\partial u^{\lceil\gamma\rceil}(x,\tau)}{\partial\tau^{\lceil\gamma\rceil}}d\tau \approx [({}_{t_{l-1},t_l}\mathbf{D}^{\gamma})^{\mathsf{T}}\otimes\mathbf{I}_P]\big(\tilde{\mathbf{U}}_{M,P}^{(l)}\big)^{\dagger}.$$

The first term of Equation (3.4), which is the memory term, approximates the hereditary effects of the fractional differential operator, accumulating the contributions from all previous subintervals $[t_{e-1},t_e]$, for $e=1,\ldots,l-1$. This term is approximated, for $t_i^e\in[t_{e-1},t_e]$, as

$$\begin{aligned}(3.5)\quad {}_{t_{e-1}}^{C}D_{t_e}^{\gamma}u(x,t) &= \frac{1}{\Gamma(\lceil\gamma\rceil-\gamma)}\int_{t_{e-1}}^{t_e}(t-\tau)^{\lceil\gamma\rceil-\gamma-1}\frac{\partial u^{\lceil\gamma\rceil}(x,\tau)}{\partial\tau^{\lceil\gamma\rceil}}d\tau\\ &\approx \sum_{n=0}^{N}\sum_{j=0}^{M}\frac{1}{\nu_j}\sum_{i=0}^{M}U_{M,N}(x_n,t_i^e)\Pi_{h,j}(t_i^e)\;\varrho_i\;{}_{t_{e-1}}^{C}D_{t_e}^{\gamma}\Pi_{h,j}(t)\mathcal{L}_n(x_e).\end{aligned}$$

When Equation (3.5) is collocated at the points $(x_e,t_r^l)\in([a,b]\times[t_{l-1},t_l])$, the derivative ${}_{t_{e-1}}^{C}D_{t_e}^{\gamma}\Pi_{h,j}(t)$ is approximated as ${}_{t_{e-1}}^{C}D_{t_e}^{\gamma}\Pi_{h_l,j}(t_r^l)\approx\boldsymbol{\Delta}\boldsymbol{\Theta}$, where the entries of $\boldsymbol{\Delta}$ and $\boldsymbol{\Theta}$ are defined as

$$\Delta_{j,k}=\begin{cases}1 & j=0,k=0\\ \frac{j(-1)^{j-k}(j+k-1)!2^{2j}}{(j-k)!(2k)!h^k} & j=1(1)M,\ k=0(1)j\\ 0 & \text{otherwise}\end{cases},$$

$$\Theta_{j,r}=\begin{cases}0 & j<\lceil\gamma\rceil\\ \dfrac{j!}{\Gamma(\lceil\gamma\rceil-\gamma)(j-\lceil\gamma\rceil)!}\displaystyle\int_{t_{e-1}}^{t_e}(t_r^l-\tau)^{\lceil\gamma\rceil-1-\gamma}(\tau-t_{e-1})^{j-\lceil\gamma\rceil}d\tau, & {\scriptstyle t_r^l\in[t_l,t_{l+1}]\atop\scriptstyle j=\lceil\gamma\rceil(1)M,\ r=0(1)M}.\end{cases}$$

Therefore, the first term of Equation (3.4) is then expressed as

$$\frac{1}{\Gamma(\lceil\gamma\rceil-\gamma)}\int_{t_{e-1}}^{t_e}(t-\tau)^{\lceil\gamma\rceil-\gamma-1}\frac{\partial u^{\lceil\gamma\rceil}(x,\tau)}{\partial\tau^{\lceil\gamma\rceil}}d\tau \approx [({}_{t_{e-1},t_e}\mathbf{D}^{\gamma})^{\mathsf{T}}\otimes\mathbf{I}_P]\big(\tilde{\mathbf{U}}_{M,P}^{(e)}\big)^{\dagger}.$$

Then, the approximation for the fractional derivative of $u(x,t)$ in the domain $([a,b]\times[t_{l-1},t_l])$ is obtained as

$$(3.6)\quad {}_{0}^{C}D_{t}^{\gamma}u(x,t)\approx\sum_{e=1}^{l-1}\big[[({}_{t_{e-1},t_e}\mathbf{D}^{\gamma})^{\mathsf{T}}\otimes\mathbf{I}_P]\big(\tilde{\mathbf{U}}_{M,P}^{(e)}\big)^{\dagger}\big]+[({}_{t_{l-1},t_l}\mathbf{D}^{\gamma})^{\mathsf{T}}\otimes\mathbf{I}_P]\big(\tilde{\mathbf{U}}_{M,P}^{(l)}\big)^{\dagger}.$$

Where $\mathbf{D}^{\gamma}$ is the $S\times S$ global differentiation matrix defined in Appendix E. This completes the proof. □

3.4. Discretization of the time-fractional PDE. Consider a general time fractional diffusion equation

$$(3.7)\qquad {}_{0}^{C}D_{t}^{\gamma}u(x,t)=\lambda\frac{\partial^2u(x,t)}{\partial x^2}+f(x,t),\quad x\in[a,b],\ t\in[0,h],\ \gamma\in(0,1],$$

where λ is the diffusion term and $f(x,t)$ is the source term, with initial condition

$$u(x,0) = g(x), \quad \text{for} \quad x \in [a,b],$$

and Dirichlet boundary conditions

$$u(a,t) = g_a(t), \quad \text{and} \quad u(b,t) = g_b(t), \quad \text{for} \quad t \in [0,h].$$

Substituting Equation (3.6) into the original PDE (3.7) yields a system of algebraic equations for each time subdomain $h_l = [t_{l-1}, t_l]$

$$\sum_{e=1}^{l-1} \Big[\big[({}_{t_{e-1},t_e}\mathbf{D}^{\gamma})^{\mathsf{T}} \otimes \mathbf{I}_P \big] \big(\tilde{\mathbf{U}}_{M,P}^{(e)} \big)^{\dagger} \Big] + \big[\big({}_{t_{l-1},t_l}\mathbf{D}^{\gamma} \big)^{\mathsf{T}} \otimes \mathbf{I}_P \big] \big(\tilde{\mathbf{U}}_{M,P}^{(l)} \big)^{\dagger} = \lambda [\mathbf{E} \otimes \mathbf{D}^2] \big(\tilde{\mathbf{U}}_{M,P}^{(l)} \big)^{\dagger} + (\mathbf{F}^{(l)})^{\dagger}, \tag{3.8}$$

where $\mathbf{F}^{(l)}$ is the vector of the source term evaluated at all grid points in the l-th subdomain. Rearranging Equation (3.8), we have

$$[[({}_{t_{l-1},t_l}\mathbf{D}^{\gamma})^{\mathsf{T}} \otimes \mathbf{I}_P] - \lambda[\mathbf{E} \otimes \mathbf{D}^2]] \big(\tilde{\mathbf{U}}_{M,P}^{(l)} \big)^{\dagger} = (\mathbf{F}^{(l)})^{\dagger} - \sum_{e=1}^{l-1} \Big[\big[({}_{t_{e-1},t_e}\mathbf{D}^{\gamma})^{\mathsf{T}} \otimes \mathbf{I}_P \big] \big(\tilde{\mathbf{U}}_{M,P}^{(e)} \big)^{\dagger} \Big]. \tag{3.9}$$

For any given l-th subdomain, the system of equations derived from Equation (3.9) can be written in matrix-vector form as

$$\mathbf{A}^{(l)} \big(\tilde{\mathbf{U}}_{M,P}^{(l)} \big)^{\dagger} = \mathbf{H}^{(l-1)}, \tag{3.10}$$

where $\mathbf{A}^{(l)} = [[({}_{t_{l-1},t_l}\mathbf{D}^{\gamma})^{\mathsf{T}} \otimes \mathbf{I}_P] - \lambda[\mathbf{E} \otimes \mathbf{D}^2]]$ and $\mathbf{H}$ is the known right-hand side constructed from previous time steps. For the first time, subdomain $l = 1$, the memory summation terms on the right-hand side are absent, and we obtain

$$\mathbf{A}^{(1)} \big(\tilde{\mathbf{U}}_{M,P}^{(1)} \big)^{\dagger} = (\mathbf{F}^{(1)})^{\dagger},$$

where $A^{(1)}$ is the system matrix from Equation (3.10) evaluated for the interval $[t_0, t_1]$, and the right-hand side is simply the source term, as the memory sum is zero.

3.4.1. *Implementation of initial and boundary conditions.* The decomposition of the temporal domain into non-overlapping suggests that

$$\tilde{\mathbf{U}}^{(l-1)}(\hat{x}_e, t_M) = \tilde{\mathbf{U}}^{(l)}(\hat{x}_e, t_0), \quad l = 1, \dots q.$$

The initial condition $u(x,0) = g(x)$ is a boundary in the time domain. The solution vector for the first subdomain, $(\tilde{\mathbf{U}}_{M,P}^{(1)})^{\dagger}$, contains the unknowns for all P spatial points at all $M+1$ time points in the subdomain. Therefore, for $t_0 = 0$, the solution vector becomes

$$\tilde{\mathbf{U}}(\hat{x}_e, t_0) = g(\hat{x}_e), \quad \text{for} \quad e = 1, 2, \dots, P.$$

For $u(a,t) = g_a(t)$, the rows of the system Equation (3.10) that correspond to the first spatial point, $\hat{x}_1 = a$, for each time node t_r within the current subdomain

h_l, are replaced by the algebraic enforcement of the boundary condition,

$$\tilde{\mathbf{U}}(\hat{x}_0, t_r) = g_a(t_r).$$

For $u(b,t) = g_b(t)$, the rows of the system Equation (3.10) that correspond to the last spatial point, $\hat{x}_P = b$, for each time node t_r within the current subdomain h_l, are replaced by the algebraic enforcement of the boundary condition

$$\tilde{\mathbf{U}}(\hat{x}_P, t_r) = g_b(t_r).$$

Enforcing these conditions transforms the system Equation (3.10) into a non-singular system, denoted as

$$\tilde{\mathbf{A}}^{(l)} \big(\tilde{\mathbf{U}}^{(l)}_{M,P}\big)^{\dagger} = \tilde{\mathbf{H}}^{(l-1)}.$$

This modified system can now be solved uniquely for the unknown solution values within the interior of the l-th subdomain. This procedure is then repeated, for $l = 2, 3, \ldots, q$.

4. Application to signal smoothing models

This section presents the application of the pseudospectral method to the signal smoothing model. For the purpose of this study, we consider linear and nonlinear models with four different clean signals.

4.1. Clean signals. The clean signals used for both linear and nonlinear models are as follows:

(1) Two-frequency sinusoidal mixture

$$\sin(2\pi x) + 0.5\sin(6\pi x).$$

(2) Single-frequency sinusoidal signal

$$\sin(2\pi x).$$

(3) Quadratic polynomial signal

$$x(x-1).$$

(4) Gaussian pulse

$$e^{-100(x-0.5)^2}.$$

4.2. Evaluation Metrics. To assess the performance of the smoothing models, three standard quantitative metrics are used: the root-mean-square error (RMSE), the signal-to-noise ratio (SNR), and the SNR gain, where *cs* represents the clean/noise-free signal and *ss* represents the smoothed signal. The formulas used to compute these metrics are outlined as follows:

(1) The RMSE measures the deviation between the smoothed signal u_{ss} and the clean signal u_{cs}, and is defined by

$$\text{RMSE} = \sqrt{\frac{1}{P}\sum_{i=1}^{P}(u_{\text{ss}}(x_i) - u_{\text{cs}}(x_i))^2}.$$

(2) The SNR quantifies the relative strength of the clean signal compared to the error and is computed as

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\sum_{i=1}^{P} u_{\mathrm{cs}}(x_i)^2}{\sum_{i=1}^{P}(u_{\mathrm{ss}}(x_i) - u_{\mathrm{cs}}(x_i))^2}\right).$$

(3) The SNR gain measures the improvement achieved by the smoothing model and is defined by

$$\mathrm{SNR}_{\mathrm{gain}} = \mathrm{SNR}_{\mathrm{out}} - \mathrm{SNR}_{\mathrm{in}},$$

where $\mathrm{SNR}_{\mathrm{in}}$ is the SNR of the noisy input signal and $\mathrm{SNR}_{\mathrm{out}}$ is the SNR of the smoothed signal.

EXAMPLE 4.1. Consider the linear time-fractional order model given by

$$\frac{\partial^{\gamma} u(x,t)}{\partial t^{\gamma}} = \lambda \frac{\partial^2 u(x,t)}{\partial x^2}, \qquad x \in [0,1] \quad \text{and} \quad \gamma \in (0,1),$$

where $u(x,t)$ is the smoothed signal, and $\lambda > 0$ is a regularization parameter.

TABLE 1. Output SNR, SNR gain, and RMSE for different signals and smoothing methods for Example 4.1, with $N = 30$, $M = 10$, $T = 10$, $q = 50$, and $p = 10$, for different values of γ, λ, and input SNR.

$\gamma = 0.7,\ \lambda = 1 \times 10^{-4},\ \mathrm{SNR} = 20$												
	Signal 1			Signal 2			Signal 3			Signal 4		
Method	$\mathrm{SNR}_{\mathrm{out}}$	$\mathrm{SNR}_{\mathrm{gain}}$	RMSE	$\mathrm{SNR}_{\mathrm{out}}$	$\mathrm{SNR}_{\mathrm{gain}}$	RMSE	$\mathrm{SNR}_{\mathrm{out}}$	$\mathrm{SNR}_{\mathrm{gain}}$	RMSE	$\mathrm{SNR}_{\mathrm{out}}$	$\mathrm{SNR}_{\mathrm{gain}}$	RMSE
TFOM	22.44	+3.12	5.96×10^{-02}	32.83	+13.52	1.61×10^{-02}	36.47	+17.16	2.74×10^{-03}	21.72	+2.41	2.90×10^{-02}
GAU	29.66	+10.35	2.59×10^{-02}	30.26	+10.95	2.17×10^{-02}	30.60	+11.29	5.38×10^{-03}	29.73	+10.42	1.15×10^{-02}
MA	28.16	+8.85	3.08×10^{-02}	30.69	+11.38	2.06×10^{-02}	32.29	+12.98	4.43×10^{-03}	28.78	+9.47	1.29×10^{-02}
SGM	28.51	+9.20	2.96×10^{-02}	28.54	+9.23	2.63×10^{-02}	28.55	+9.24	6.81×10^{-03}	28.53	+9.22	1.32×10^{-02}

$\gamma = 0.9,\ \lambda = 1 \times 10^{-6},\ \mathrm{SNR} = 30$												
	Signal 1			Signal 2			Signal 3			Signal 4		
Method	$\mathrm{SNR}_{\mathrm{out}}$	$\mathrm{SNR}_{\mathrm{gain}}$	RMSE	$\mathrm{SNR}_{\mathrm{out}}$	$\mathrm{SNR}_{\mathrm{gain}}$	RMSE	$\mathrm{SNR}_{\mathrm{out}}$	$\mathrm{SNR}_{\mathrm{gain}}$	RMSE	$\mathrm{SNR}_{\mathrm{out}}$	$\mathrm{SNR}_{\mathrm{gain}}$	RMSE
TFOM	38.62	+9.31	9.25×10^{-03}	38.53	+9.22	8.36×10^{-03}	38.53	+9.22	2.16×10^{-03}	38.40	+9.09	4.25×10^{-03}
GAU	35.51	+6.20	1.32×10^{-02}	38.35	+9.04	8.53×10^{-03}	40.33	+11.02	1.75×10^{-03}	36.01	+6.70	5.59×10^{-03}
MA	30.03	+0.72	2.49×10^{-02}	34.80	+5.49	1.28×10^{-02}	39.60	+10.29	1.91×10^{-03}	31.33	+2.02	9.58×10^{-03}
SGM	38.14	+8.83	9.78×10^{-03}	38.42	+9.11	8.47×10^{-03}	38.52	+9.21	2.16×10^{-03}	38.12	+8.81	4.39×10^{-03}

Table 1 presents the output SNR, SNR gain, and RMSE obtained for four test signals processed using the time-fractional order model (TFOM), Gaussian (GAU), Moving Average (MA), and Savitzky-Golay (SGM) smoothing methods for Example 4.1. The experiments were conducted under two parameter settings. The first corresponds to a moderately fractional model with stronger regularization $\gamma = 0.7$, $\lambda = 1 \times 10^{-4}$, and a noisy input $SNR = 20$. The second reflects a higher fractional model and weaker regularization $\gamma = 0.9$, $\lambda = 1 \times 10^{-6}$, and a noisy input $SNR = 30$. For $SNR = 20$, TFOM consistently produces the lowest RMSE values across all four signals, demonstrating robustness in suppressing noise while maintaining the underlying signal features. The TFOM method achieves substantial SNR improvements, particularly for Signals 2 and 3, where the SNR gains exceed 13 dB and 17 dB, respectively. The GAU and MA methods also yield measurable improvements but generally exhibit higher reconstruction errors than TFOM, with MA displaying more variability in performance. The SGM method yields stable yet modest gains, with SNR improvements of around 9 dB for most signals. When

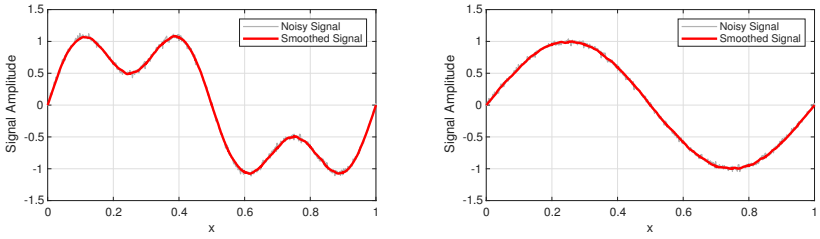
the noise level is $SNR = 30$, all methods show improved output SNR and reduced RMSE. TFOM maintains its advantage by achieving consistently low RMSE values across all signals, confirming its accuracy even under weak regularization. GAU and SGM perform comparably in this parameter setting, delivering high output SNRs and reasonable RMSE values. In contrast, MA becomes more sensitive to the reduced regularization, leading to larger errors for some signal cases. Overall, the results indicate that TFOM exhibits stability across different fractional-orders, regularization parameters, and noise levels. Its consistently low RMSE and competitive SNR performance highlight its effectiveness in balancing noise suppression and signal fidelity, making it a reliable method for a wide range of smoothing applications.

Table 2 compares the performance of four smoothing methods, namely, time-fractional order model (TFOM) solved using the pseudospectral multidomain method, Gaussian (GAU), Moving Average (MA), and Savitzky-Golay (SGM), applied to four different signals under three noise levels SNR=10, 20, and 30 dB for Example 4.1. Across all experiments, the proposed TFOM method consistently achieves the best overall performance, producing the highest output SNR and the lowest RMSE in almost every case. Notably, TFOM maintains a stable SNR gain of approximately +12 dB regardless of the input noise level or signal type, demonstrating strong robustness to noise variability. The MA filter performs competitively at a low SNR (10 dB), sometimes exceeding the TFOM in SNR gain; however, its performance deteriorates significantly as the input SNR increases, indicating over-smoothing when the noise is weak. The GAU and SGM methods provide moderate and relatively stable improvements, although their SNR gains remain lower than those of TFOM, particularly at higher noise levels. RMSE values decrease for all methods as the input SNR increases, with TFOM achieving the smallest values, reaching as low as 1×10^{-03} for SNR=30 dB. Overall, the results clearly show that TFOM provides reliable and accurate smoothing across different noise conditions and signal types, outperforming the three smoothing methods compared.

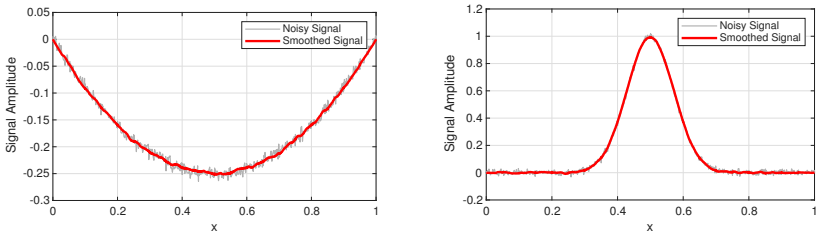
Figure 1 presents the performance of the proposed TFOM smoothing method on four different types of signals, each with distinct structural characteristics, for Example 4.1. In all cases, the noisy input contains high-frequency disturbances superimposed on the true underlying profile, while the TFOM-smoothed output closely reconstructs the clean signal. Overall, the four subfigures collectively demonstrate the robustness and adaptability of the TFOM model. Across all four types of signals, the method consistently achieves a strong balance between detail preservation and noise removal, validating its suitability for a wide range of one-dimensional signal-processing tasks. Figure 2 represents a comparative analysis of different signal smoothing methods for Example 4.1. The subfigures demonstrate that the TFOM algorithm is robust and capable of reconstructing various signal types with an accuracy that matches or rivals that of standard methods, such as Savitzky-Golay and Gaussian smoothing. Figure 3 presents the RMSE comparison for four different signal smoothing methods applied to various signal types for Example 4.1. The subfigures serve as the statistical proof for the study. They demonstrate that TFOM is the most robust method among those tested. It adapts best to different

TABLE 2. Output SNR, SNR gain, and RMSE for different signals for different smoothing methods for Example 4.1, for $N = 30$, $M = 10$, $T = 10$, $\gamma = 0.5$, $\lambda = 1 \times 10^{-5}$, $q = 20$, and $p = 20$.

Signal	SNR	Smoothing Method	SNR_{in}	SNR_{out}	SNR_{gain}	RMSE
1	10	TFOM	9.31	21.47	+12.16	6.66×10^{-02}
		GAU		20.40	+20.40	7.54×10^{-02}
		MA		21.62	+12.31	6.55×10^{-02}
		SGM		18.55	+9.24	9.33×10^{-02}
	20	TFOM	19.31	31.58	+12.27	2.08×10^{-02}
		GAU		20.40	+10.35	2.59×10^{-02}
		MA		21.62	+8.85	3.08×10^{-02}
		SGM		18.55	+9.20	2.96×10^{-02}
	30	TFOM	29.31	40.73	+11.43	7.25×10^{-03}
		GAU		35.51	+6.20	1.32×10^{-02}
		MA		30.03	+0.72	2.49×10^{-02}
		SGM		38.14	+8.82	9.78×10^{-03}
2	10	TFOM	9.31	21.37	+12.06	6.03×10^{-02}
		GAU		20.47	+11.16	6.69×10^{-02}
		MA		22.09	+12.78	5.55×10^{-02}
		SGM		18.55	+9.24	8.34×10^{-02}
	20	TFOM	19.31	31.38	+12.07	1.91×10^{-02}
		GAU		30.26	+10.95	2.17×10^{-02}
		MA		30.69	+11.38	2.06×10^{-02}
		SGM		28.54	+9.23	2.64×10^{-02}
	30	TFOM	29.31	41.33	+12.02	6.06×10^{-03}
		GAU		38.35	+9.04	8.53×10^{-03}
		MA		34.80	+5.49	1.28×10^{-02}
		SGM		38.42	+9.11	8.47×10^{-03}
3	10	TFOM	9.31	21.36	+12.05	1.56×10^{-02}
		GAU		20.53	+11.22	1.71×10^{-02}
		MA		22.37	+13.06	1.39×10^{-02}
		SGM		18.55	+9.24	2.15×10^{-02}
	20	TFOM	19.31	31.35	+12.04	4.93×10^{-03}
		GAU		30.60	+11.29	5.38×10^{-03}
		MA		32.29	+12.98	4.43×10^{-03}
		SGM		28.55	+9.24	6.81×10^{-03}
	30	TFOM	29.31	41.33	+12.02	1.56×10^{-03}
		GAU		40.33	+11.02	1.75×10^{-03}
		MA		39.60	+10.29	1.91×10^{-03}
		SGM		38.52	+9.21	2.16×10^{-03}
4	10	TFOM	9.31	21.29	+11.98	3.05×10^{-02}
		GAU		20.38	+11.07	3.38×10^{-02}
		MA		21.63	+12.32	2.93×10^{-02}
		SGM		18.55	+9.24	4.18×10^{-02}
	20	TFOM	19.31	31.00	+11.69	9.96×10^{-03}
		GAU		29.73	+10.42	1.15×10^{-02}
		MA		28.78	+9.47	1.29×10^{-02}
		SGM		28.53	+9.22	1.32×10^{-02}
	30	TFOM	29.31	39.26	+9.95	3.85×10^{-03}
		GAU		36.01	+6.70	5.59×10^{-03}
		MA		31.33	+2.02	9.58×10^{-03}
		SGM		38.12	+8.81	4.39×10^{-03}

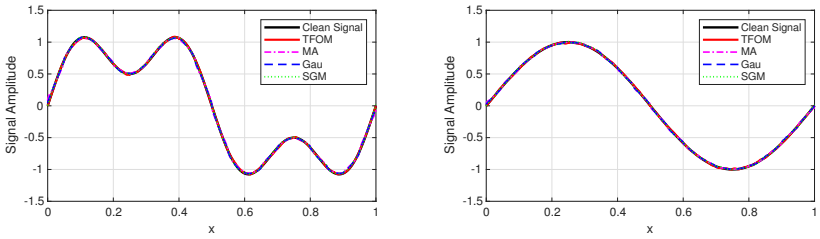


(A) (B)

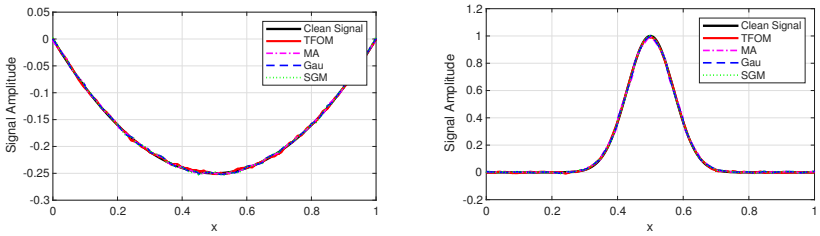


(C) (D)

FIGURE 1. Noisy signal and smoothed signal for Example 4.1 with $\gamma = 0.5$, $\lambda = 1\times10^{-5}$, SNR $= 30$, $q = 20$, $p = 20$, $N = 30$, $M = 10$, and $T = 10$.



(A) (B)



(C) (D)

FIGURE 2. Comparison of different signal smoothing methods for Example 4.1 with $\gamma = 0.5$, $\lambda = 1 \times 10^{-5}$, SNR $= 30$, $q = 20$, $p = 20$, $N = 30$, $M = 10$, and $T = 10$.

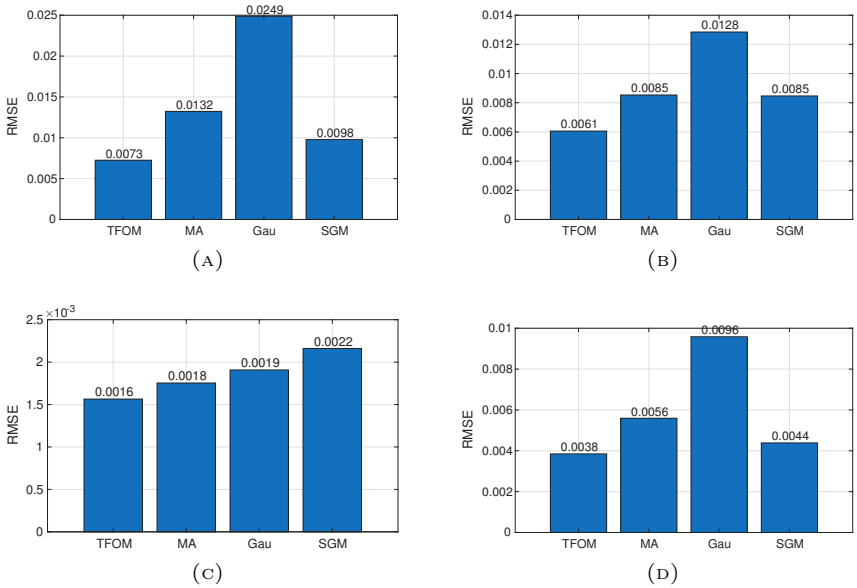


FIGURE 3. RMSE for Example 4.1 with $\gamma = 0.5$, $\lambda = 1 \times 10^{-5}$, SNR $= 30$, $q = 20$, $p = 20$, $N = 30$, $M = 10$, and $T = 10$.

signal shapes such as peaks, curves, and complex waves, without sacrificing accuracy, whereas traditional methods like Gaussian smoothing struggle to maintain signal fidelity in complex scenarios.

EXAMPLE 4.2. The Rudin-Osher-Fatemi (ROF) model is a classical and widely used approach for edge-preserving signal and image smoothing. It originates from the total variation (TV) technique introduced by Rudin et al. [25], which aims to remove noise while retaining sharp transitions and important structural features. In one spatial dimension, the model yields a nonlinear diffusion equation in which the diffusion strength decreases near sharp gradients, thereby preventing excessive smoothing of edges. Consider the time-fractional version of the ROF diffusion equation, given by

$$\frac{\partial^\gamma u(x,t)}{\partial t^\gamma} = \frac{\partial g(u_x)}{\partial x}\, u_x + g(u_x)\, u_{xx}, \quad x \in [0,1],\ \gamma \in (0,1),$$

where $u(x,t)$ represents the smoothed signal at time t, and γ is the Caputo fractional order controlling the memory effect. The function

$$g(u_x) = \frac{1}{\sqrt{u_x^2 + \lambda^2}}$$

is the nonlinear diffusivity, with $\lambda > 0$ being a regularization parameter that prevents division by zero and controls the sensitivity of the diffusion process to sharp gradients. The term $g(u_x)$ decreases as $|u_x|$ increases, thereby reducing diffusion across steep transitions and preserving edges. Conversely, when the gradient is small, $g(u_x)$ becomes larger, allowing stronger smoothing in relatively flat regions.

This mechanism is a hallmark of ROF-type models, enabling selective, adaptive noise removal. The nonlinear terms involving $g(u_x)$ and its derivative are handled using the quasilinearization strategy introduced by Bellman and Kalaba [26]. This ensures a stable and efficient numerical implementation suitable for both sharp and smooth signal features.

Table 3 compares the performance of four smoothing techniques, ROF solved using the pseudospectral multidomain method, GAU, MA, and SGM, on the Example 4.2, applied to four signals under varying input SNR levels 10, 20, and 30 dB, using output SNR, SNR gain, and RMSE as metrics. The ROF consistently achieves the highest or near-highest output SNR and the lowest RMSE across all cases, demonstrating robust denoising even under heavy noise conditions. At 10 dB, ROF gains range from +12 to +14 dB, while at 20 dB they remain strong, ranging from +8 to +13 dB. At 30 dB, moderate improvement is still observed, with approximately +10 dB for Signals 2 and 3. Gaussian smoothing performs reasonably but is consistently outperformed by ROF, with gains decreasing at higher SNR and occasional over-blurring that increases RMSE. The MA filter exhibits the weakest performance, often producing low or negative gains and high RMSE due to its inability to preserve edges or nonlinear features. SGM performs competitively at high SNR, sometimes exceeding ROF, but its effectiveness declines under heavy noise, with gains around +9 dB at 10 dB SNR. Overall, ROF provides superior noise reduction, better preservation of signal structure, and lower reconstruction error, with its advantage most pronounced at low SNR, highlighting the efficacy of fractional-order operators in modeling and suppressing noise in complex signals.

Figure 4 illustrates the performance of the Rudin-Osher-Fatemi (ROF) diffusion model on four different types of signals, each exhibiting distinct structural characteristics. In all cases, the noisy inputs contain high-frequency disturbances superimposed on the true underlying signal, while the ROF-smoothed outputs effectively reconstruct the clean profiles. The results demonstrate that the ROF model is robust and adaptable, consistently achieving a strong balance between noise suppression and edge preservation. Across various signal types, the method preserves sharp transitions and fine structures without oversmoothing, confirming its suitability for nonlinear and complex signal processing tasks.

Figure 5 presents a comparative analysis of the ROF model against classical smoothing methods, including GAU, MA, and SGM filtering. The subfigures show that the ROF algorithm achieves comparable reconstruction accuracy across all signals.

Figure 6 presents the RMSE comparison for all four smoothing methods across the different signal types. These results quantitatively support the robustness of the ROF model; it adapts well to a variety of signal shapes while maintaining a low reconstruction error. In contrast, standard methods such as Gaussian smoothing exhibit higher RMSE and reduced accuracy when processing signals with pronounced features or nonlinearities. Overall, these figures confirm that the ROF diffusion model provides effective denoising while preserving essential signal characteristics, making it particularly suitable for nonlinear signals.

TABLE 3. Output SNR, SNR gain, and RMSE for different signals for different smoothing methods for Example 4.2, with $N = 50$, $M = 8$, $T = 2$, $\gamma = 0.85$, $\lambda = 10 \times 10^3$, $q = 5$, and $p = 10$.

Signal	SNR	Smoothing Method	SNR_{in}	SNR_{out}	SNR_{gain}	RMSE
1	10	ROF	10.05	23.76	+13.70	5.11×10^{-02}
		GAU		20.44	+10.39	7.49×10^{-02}
		MA		19.01	+8.96	8.82×10^{-02}
		SGM		19.41	+9.36	8.43×0^{-02}
	20	ROF	20.05	30.21	+10.16	2.43×10^{-02}
		GAU		26.61	+6.56	3.68×10^{-02}
		MA		21.15	+1.06	9.90×10^{-02}
		SGM		29.19	+9.14	2.73×10^{-02}
	30	ROF	30.05	30.90	+0.84	2.25×10^{-02}
		GAU		28.28	−1.78	3.04×10^{-02}
		MA		21.41	−8.64	6.69×10^{-02}
		SGM		38.07	+8.01	9.83×10^{-03}
2	10	ROF	10.05	23.39	+13.33	4.77×10^{-02}
		GAU		21.26	+11.20	6.09×10^{-02}
		MA		22.04	+11.98	5.57×10^{-02}
		SGM		19.49	+9.43	7.47×10^{-02}
	20	ROF	20.05	33.02	+12.96	1.57×10^{-02}
		GAU		30.50	+10.45	2.10×10^{-02}
		MA		27.87	+7.81	2.85×10^{-02}
		SGM		29.48	+9.43	2.36×10^{-02}
	30	ROF	30.05	40.12	+10.06	6.95×10^{-03}
		GAU		35.52	+5.47	1.18×10^{-02}
		MA		28.99	−1.06	2.50×10^{-02}
		SGM		39.37	+9.32	7.57×10^{-03}
3	10	ROF	10.05	23.41	+13.36	1.23×10^{-02}
		GAU		21.10	+11.05	1.60×10^{-02}
		MA		22.19	+12.14	1.41×10^{-02}
		SGM		19.47	+9.42	1.93×10^{-02}
	20	ROF	20.05	33.42	+13.37	3.88×10^{-03}
		GAU		30.69	+10.64	5.31×10^{-03}
		MA		30.41	+10.35	5.49×10^{-03}
		SGM		29.45	+9.39	6.13×10^{-03}
	30	ROF	30.05	43.22	+13.16	1.26×10^{-03}
		GAU		38.44	+8.39	2.17×10^{-03}
		MA		34.25	+4.19	3.53×10^{-03}
		SGM		39.35	+9.29	1.96×10^{-03}
4	10	ROF	10.05	22.77	+12.72	2.34×10^{-02}
		GAU		20.95	+10.90	2.89×10^{-02}
		MA		19.27	+9.21	3.50×10^{-02}
		SGM		19.59	+9.54	3.37×10^{-02}
	20	ROF	20.05	28.71	+8.65	1.18×10^{-02}
		GAU		26.85	+6.80	1.46×10^{-02}
		MA		21.05	+1.30	2.83×10^{-02}
		SGM		29.62	+9.56	1.06×10^{-02}
	30	ROF	30.05	30.02	−0.03	1.02×10^{-02}
		GAU		27.82	−2.23	1.31×10^{-02}
		MA		21.20	−8.86	2.80×10^{-02}
		SGM		38.05	+7.99	4.03×10^{-03}

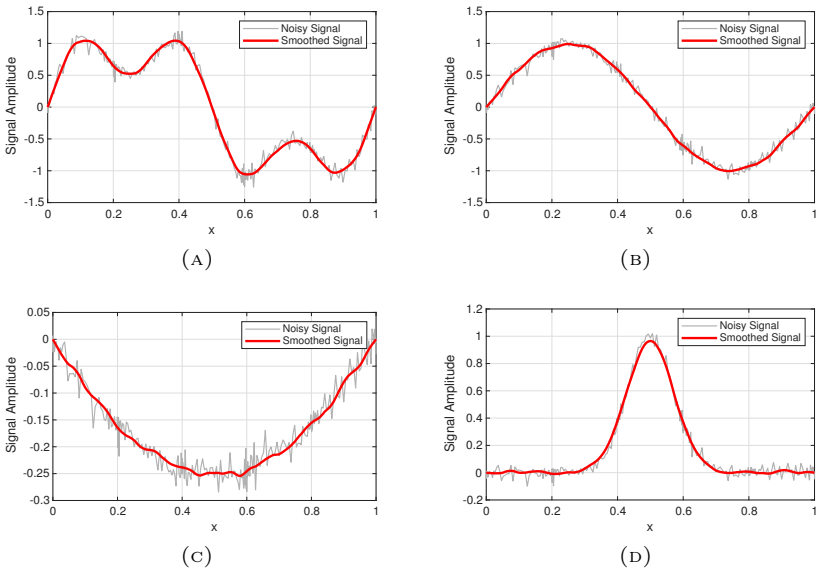


FIGURE 4. Noisy signal and smoothed signal for Example 4.2 with $\gamma = 0.85$, $\lambda = 10 \times 10^3$, SNR $= 20$, $q = 5$, $p = 10$, $N = 50$, $M = 8$, and $T = 2$.

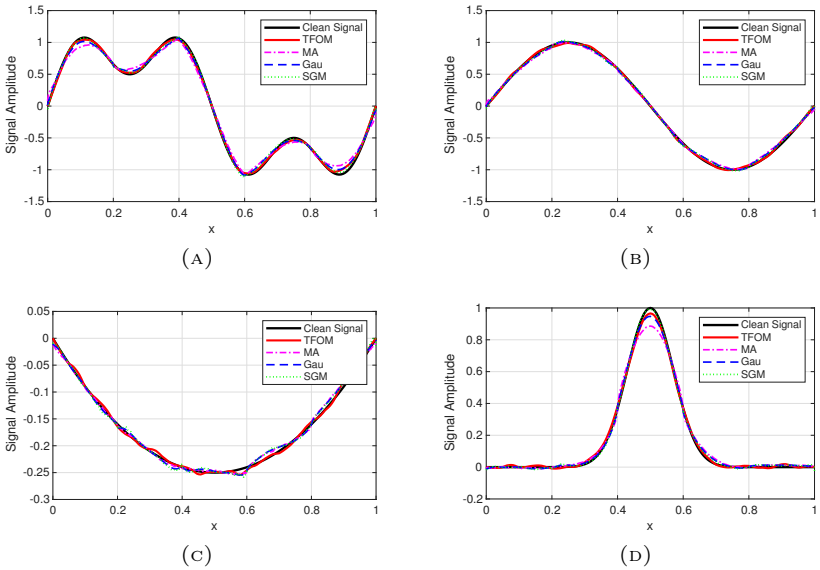


FIGURE 5. Comparison of different signal smoothing methods for Example 4.2 with $\gamma = 0.85$, $\lambda = 10 \times 10^3$, SNR $= 20$, $q = 5$, $p = 10$, $N = 50$, $M = 8$, and $T = 2$.

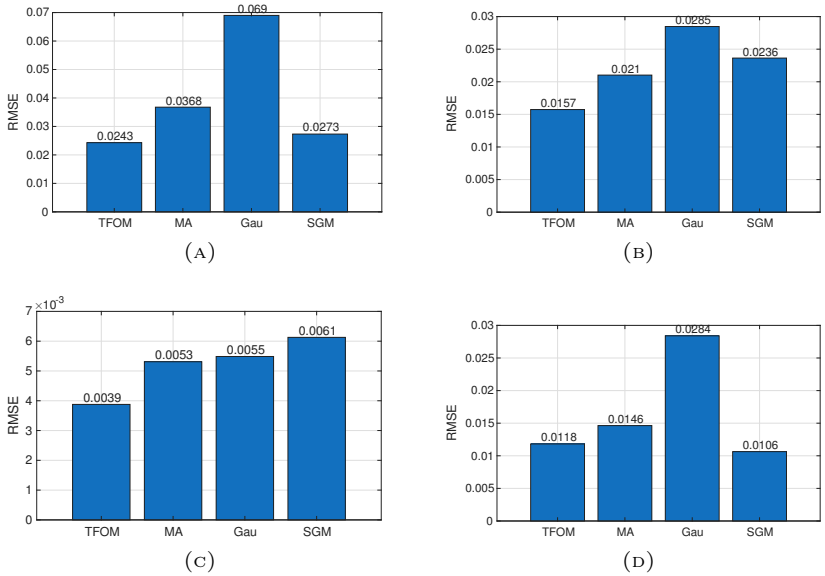


FIGURE 6. RMSE for Example 4.2 with $\gamma = 0.85$, $\lambda = 10 \times 10^3$, SNR $= 20$, $q = 5$, $p = 10$, $N = 50$, $M = 8$, and $T = 2$.

5. Conclusion

This study presented the results of applying advanced smoothing techniques to both linear and nonlinear one-dimensional models. The multidomain pseudospectral method proposed was developed by discretizing the spatial domain using Lagrange interpolating polynomials and the temporal domain using shifted Chebyshev polynomials of the first kind. The spatial domain was decomposed into equal overlapping subdomains, whereas the temporal domain was decomposed into equal non-overlapping subdomains. For the linear model, the time-fractional order model (TFOM) was used, demonstrating consistently high output SNR and low RMSE across varying noise levels. The TFOM effectively balances noise reduction with preservation of signal details, outperforming classical methods such as Gaussian, Moving Average, and Savitzky-Golay smoothing methods, particularly under low SNR conditions.

For the nonlinear model, the Rudin-Osher-Fatemi (ROF) diffusion model was applied. The ROF model effectively preserves edges and structural features while suppressing noise, making it particularly suitable for signals with sharp transitions or complex nonlinear characteristics. Comparative analysis with classical methods confirmed that ROF maintains lower reconstruction error and higher accuracy across a range of signal types.

Both the TFOM and ROF models were solved using the pseudospectral multidomain method, which enables accurate and efficient numerical approximation of derivatives and fractional-order operators. The pseudospectral multidomain

method enhances the stability and robustness of the models, ensuring reliable reconstruction across different signal types and noise levels.

Overall, the results confirm that the pseudospectral multidomain method effectively captures signal characteristics: the TFOM is optimal for linear signals, while the ROF model is preferable for nonlinear signals. In all cases, the combination of advanced modeling and the pseudospectral multidomain numerical approach provides superior noise reduction, accurate reconstruction, and structural preservation compared to classical smoothing techniques.

The primary objective of this paper was to develop and implement a multidomain pseudospectral method for solving time-fractional signal smoothing models and to demonstrate its effectiveness through numerical experiments. The results show that the proposed method provides accurate, stable numerical solutions for both linear and nonlinear signal-smoothing problems. A rigorous theoretical convergence and stability analysis of the proposed multidomain pseudospectral scheme is beyond the scope of the present study and will be the subject of future research.

Appendix A. The decomposition of the spatial domain into p overlapping subdomains

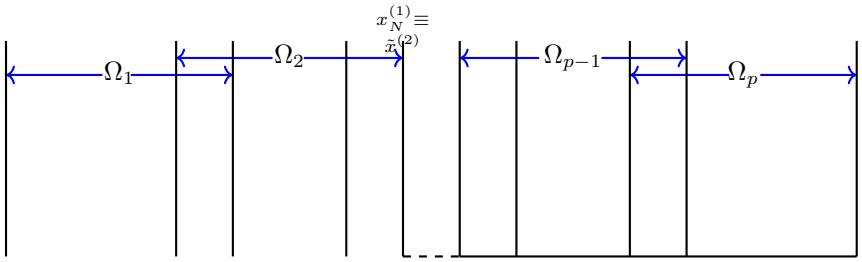


FIGURE 7. Pictorial representation of the decomposition of the spatial domain into p subdomains that overlap with one another

Appendix B. The matrix D representing the entire spatial domain is an $P \times P$ matrix stacked as follows

$$
\mathbf{D} = \begin{bmatrix}
\overset{1}{D}_{0,0} & \overset{1}{D}_{0,1} & \dots & \overset{1}{D}_{0,N-1} & \overset{1}{D}_{0,N} & & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & & & & & \\
\overset{1}{D}_{N-1,0} & \overset{1}{D}_{N-1,1} & \dots & \overset{1}{D}_{N-1,N-1} & \overset{1}{D}_{N-1,N} & & & & & \\
 & & & \overset{2}{D}_{1,0} & \overset{2}{D}_{1,1} & \dots & \overset{2}{D}_{1,N-1} & \overset{2}{D}_{1,N} & & \\
 & & & \vdots & \vdots & \vdots & \vdots & \vdots & & \\
 & & & \overset{2}{D}_{N-1,0} & \overset{2}{D}_{N-1,1} & \dots & \overset{2}{D}_{N-1,N-1} & \overset{2}{D}_{N-1,N} & & \\
 & & & & & & \ddots & \ddots & & \\
 & & & & & & \ddots & \ddots & & \\
 & & & & & & \overset{p}{D}_{1,0} & \overset{p}{D}_{1,1} & \dots & \overset{p}{D}_{1,N} \\
 & & & & & & \vdots & \vdots & \vdots & \vdots \\
 & & & & & & \overset{p}{D}_{N,0} & \overset{p}{D}_{N,1} & \dots & \overset{p}{D}_{N,N}
\end{bmatrix}.
$$

Appendix C. The partitioning of the temporal domain $[0, h]$ into p equal, non-overlapping subintervals

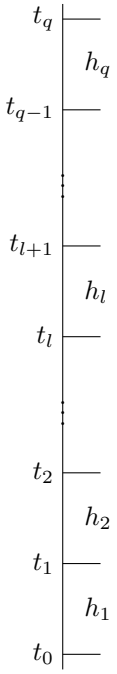


FIGURE 8. Illustrative representation of dividing $[0, h]$ into q non-overlapping equal subintervals.

Appendix D. The Complete Discretization Grid

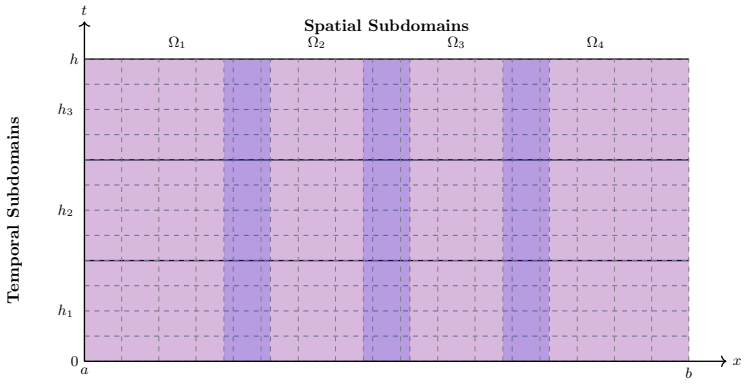


FIGURE 9. The domain $[a, b] \times [0, h]$ is partitioned into overlapping spatial subdomains Ω_l in vertical bands and non-overlapping temporal subdomains h_l in horizontal bands.

Appendix E. $S \times S$ global differentiation matrix

$$
\begin{bmatrix}
(\mathbf{D}^{\psi})^{\mathbf{I}}_{0,0} & (\mathbf{D}^{\psi})^{\mathbf{I}}_{0,1} & \cdots & (\mathbf{D}^{\psi})^{\mathbf{I}}_{0,M-1} & (\mathbf{D}^{\psi})^{\mathbf{I}}_{0,M} & & & & & & & & & & & & & & & & & & & & & & \\
(\mathbf{D}^{\psi})^{\mathbf{L}}_{1,0} & (\mathbf{D}^{\psi})^{\mathbf{L}}_{1,1} & \cdots & (\mathbf{D}^{\psi})^{\mathbf{L}}_{1,M-1} + (\mathcal{D}^{\psi})^{\mathbf{I}}_{1,0} & (\mathbf{D}^{\psi})^{\mathbf{L}}_{1,M} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{1,1} & \cdots & (\mathbf{D}^{\psi})^{\mathbf{I}}_{1,M-1} & (\mathbf{D}^{\psi})^{\mathbf{I}}_{1,M} & & & & & & & & & & & & & & & & & & \\
(\mathcal{D}^{\psi})^{\mathbf{I}}_{1,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{1,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{1,M-1} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{1,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{1,M} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{1,1} & \cdots & (\mathbf{D}^{\psi})^{\mathbf{I}}_{1,M-1} & (\mathbf{D}^{\psi})^{\mathbf{I}}_{1,M} & & & & & & & & & & & & & & & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & & & & & & & & & & & & & & & & \\
(\mathcal{D}^{\psi})^{\mathbf{I}}_{M-1,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M-1,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M-1,M-1} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{M-1,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M-1,M} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{M-1,1} & \cdots & (\mathbf{D}^{\psi})^{\mathbf{I}}_{M-1,M-1} & (\mathbf{D}^{\psi})^{\mathbf{I}}_{M-1,M} & & & & & & & & & & & & & & & & & & \\
(\mathcal{D}^{\psi})^{\mathbf{I}}_{M,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M,M-1} + (\mathcal{D}^{\psi})^{\mathbf{I}}_{M,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M,M} + (\mathcal{D}^{\psi})^{\mathbf{I}}_{M,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M,M-1} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{M,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M,M} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{M,1} & \cdots & (\mathbf{D}^{\psi})^{\mathbf{I}}_{M,M-1} & (\mathbf{D}^{\psi})^{\mathbf{I}}_{M,M} & & & & & & & & & & & & & & & & \\
(\mathcal{D}^{\psi})^{\mathbf{I}}_{M+1,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M+1,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M+1,M-1} + (\mathcal{D}^{\psi})^{\mathbf{I}}_{M+1,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M+1,M} + (\mathcal{D}^{\psi})^{\mathbf{I}}_{M+1,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M+1,M-1} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{M+1,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M+1,M} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{M+1,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M+1,M-1} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{M+1,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{M+1,M} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{M+1,1} & \cdots & (\mathbf{D}^{\psi})^{\mathbf{I}}_{M+1,M-1} & (\mathbf{D}^{\psi})^{\mathbf{I}}_{M+1,M} & & & & & & & & & & & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
(\mathcal{D}^{\psi})^{\mathbf{I}}_{S-1,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S-1,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S-1,M-1} + (\mathcal{D}^{\psi})^{\mathbf{I}}_{S-1,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S-1,M} + (\mathcal{D}^{\psi})^{\mathbf{I}}_{S-1,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S-1,M-1} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{S-1,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S-1,M} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{S-1,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S-1,M-1} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{S-1,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S-1,M} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{S-1,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S-1,M-1} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{S-1,0} & (\mathbf{D}^{\psi})^{\mathbf{I}}_{S-1,M} & & & & & & & & & & & & & & \\
(\mathcal{D}^{\psi})^{\mathbf{I}}_{S,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S,M-1} + (\mathcal{D}^{\psi})^{\mathbf{I}}_{S,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S,M} + (\mathcal{D}^{\psi})^{\mathbf{I}}_{S,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S,M-1} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{S,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S,M} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{S,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S,M-1} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{S,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S,M} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{S,1} & \cdots & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S,M-1} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{S,0} & (\mathcal{D}^{\psi})^{\mathbf{I}}_{S,M} + (\mathbf{D}^{\psi})^{\mathbf{I}}_{S,1} & \cdots & (\mathbf{D}^{\psi})^{\mathbf{I}}_{S,M-1} & (\mathbf{D}^{\psi})^{\mathbf{I}}_{S,M}
\end{bmatrix}
$$

References

1. J. A. Ansari, I. Usman, *A review of the importance of digital signal processing in the medical field*, International Journal of Computing and Related Technologies **4**(2) (2023), 1–11.
2. A. Mathur, A. Akhtar, *Fractional calculus in signal processing: The use of fractional calculus in signal processing applications, such as image denoising, filtering, and time series analysis*, International Journal for Multidisciplinary Research **5** (2023), 23045469.
3. V. E. Tarasov, *Generalized memory: Fractional calculus approach*, Fractal and Fractional **2**(4) (2018), 23.
4. D. Constantinescu, M. Stoicescu, *Fractal dynamics as long-range memory modeling technique*, Physics AUC **21** (2011), 114–120.
5. E. A. Gonzalez , I. Petráš, *Advances in fractional calculus: Control and signal processing applications*, in: *Proceedings of the 2015 16th International Carpathian Control Conference (ICCC)*, 147–152, IEEE , 2015.
6. C. I. Muresan, I. R. Birs, E. H. Dulf, D. Copot, L. Miclea, *A review of recent advances in fractional-order sensing and filtering techniques*, Sensors **21**(17) (2021), 5920.
7. M. D. Ortigueira, C. M. Ionescu, J. T. Machado, J. J. Trujillo, *Fractional signal processing and applications*, Signal Process. **107** (2015), 197.
8. M. Shams, M. A. Rufai, *Fractional-order numerical scheme with symmetric structure for fractional differential equations with step-size control*, Symmetry **17**(10) (2025), 1685.
9. Y. Li, F. Liu, I. W. Turner, T. Li, *Time-fractional diffusion equation for signal smoothing*, Appl. Math. Comput. **326** (2018), 108–116.
10. A. P. Jayaraj, K. N. Gounder, J. Rajagopal, *Optimizing signal smoothing using HERS algorithm and time fractional diffusion equation*, Expert Systems with Applications **238** (2024), 122250.
11. P. D. Hatziantoniou, J. N. Mourjopoulos, *Generalized fractional-octave smoothing of audio and acoustic responses*, Journal of the Audio Engineering Society **48**(4) (2000), 259–280.
12. Y. F. Pu, J. L. Zhou, P. Siarry, N. Zhang, Y. G. Liu, *Fractional partial differential equation: Fractional total variation and fractional steepest descent approach-based multiscale denoising model for texture image*, Abstr. Appl. Anal. **2013**(1) (2013), 483791.
13. N. He, J. B. Wang, L. L. Zhang, K. Lu, *An improved fractional-order differentiation model for image denoising*, Signal Process. **112** (2015), 180–188.
14. Y. Li, *Signal smoothing with time-space fractional order model*, Measurement Science Review **21**(1) (2021), 25–32.
15. C. Tan, L. Yu, J. Tan, Y. Chen, C. He, S. Yuan, *Transient electromagnetic signal filtering method based on intelligent optimized time-space fractional-order diffusion equation*, IEEE Access **12** (2024), 91025–91039.
16. Q. Wang, J. Ma, S. Yu, L. Tan, *Noise detection and image denoising based on fractional calculus*, Chaos Solitons Fractals **131** (2020), 109463.
17. G. Huang, H. Y. Qin, Q. Chen, Z. Shi, S. Jiang, C. Huang, *Research on application of fractional calculus operator in image underlying processing*, Fractal and Fractional **8**(1) (2024), 37.
18. J. X. Wu, P. Y. Chen, C. M. Li, Y. C. Kuo, N. S. Pai, C. H. Lin, *Multilayer fractional-order machine vision classifier for rapid typical lung diseases screening on digital chest X-ray images*, IEEE Access **8** (2020), 105886–105902.
19. C. Y. Ma, B. Shiri, G. C. Wu, D. Baleanu, *New fractional signal smoothing equations with short memory and variable order*, Optik **218** (2020), 164507.
20. T. Liu, B. Ding, B. N. Saray, D. A. Juraev, E. E. Elsayed, *On the pseudospectral method for solving the fractional Klein–Gordon equation using Legendre cardinal functions*, Fractal and Fractional **9**(3) (2025), 177.
21. S. D. Oloniiju, N. Mukwevho, Y. O. Tijani, O. Otegbeye, *Chebyshev pseudospectral method for fractional differential equations in non-overlapping partitioned domains*, AppliedMath **4**(3) (2024), 950–974.

22. L. N. Trefethen, *Is Gauss quadrature better than Clenshaw–Curtis?*, SIAM Rev. **50**(1) (2008), 67–87.
23. S. Oloniiju, N. Nkomo, S. Goqo, P. Sibanda, *Shifted Chebyshev spectral method for two-dimensional space-time fractional partial differential equations*, Preprints (2022). DOI: https://doi.org/10.20944/preprints202205.0138.v1.
24. L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM, 2000.
25. L. I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D **60**(1–4) (1992), 259–268.
26. R. E. Bellman and R. E. Kalaba, *Quasilinearization and Nonlinear Boundary-Value Problems*, Modern analytic and computational methods in science and mathematics, American Elsevier Publishing Company, 1965.

ВРЕМЕНСКИ ФРАКЦИОНИ МОДЕЛИ ЗА ГЛАЧАЊЕ СИГНАЛА ПРИМЕНОМ ВИШЕДОМЕНСКЕ ПСЕУДОСПЕКТРАЛНЕ МЕТОДЕ

Резиме. Обрада и глачање зашумљених сигнала представља фундаментални изазов у многим научним и инжењерским дисциплинама. Временски фракциони дифузиони модели су се недавно појавили као моћни алати за решавање овог изазова, нудећи значајне предности у односу на класичне методе глачања тако што обухватају ефекте меморије и боље чувају оштре карактеристике и ивице. Ова истраживање се фокусира на нумеричко решавање таквих временски фракционих модела за једнодимензионално глачање сигнала. Главни извор рачунарске тешкоће у овим моделима произилази из нелокалне природе временског Капутовог (Caputo) извода реда γ, чије језгро дуготрајне меморије захтева пажљиву и тачну апроксимацију током читаве историје времена. Како би се одговорило на ове изазове, у овом раду је развијен нови вишедоменски псеудоспектрални приступ. Конкретно, временски домен је дискретизован применом вишедоменског приступа без преклапања, док се просторним доменом управља стратегијом декомпозиције домена са преклапањем. Апроксимација функције се састоји од Лагранжових интерполационих полинома за просторни домен и помереног Чебишевљевог полинома прве врсте за временски домен. Циљ ове студије је да комбинује ове нумеричке компоненте у јединствену и робусну шему и да је по први пут примени на проблем глачања сигнала фракционог реда. Студија показује ефикасност развијене псеудоспектралне вишедоменске методе када се примени и на линеарне и на нелинеарне временски фракционе моделе глачања, укључујући Рудин–Ошер–Фатеми (ROF) дифузиони модел. Резултати су приказани графички и у табеларној форми.

Department of Mathematics
Rhodes University
Grahamstown, South Africa

Department of Mathematical and Computational Sciences
University of Venda
Thohoyandou, South Africa
nancy.mukwevho@univen.ac.za
https://orcid.org/0000-0002-9689-260X

School of Computer Science and Applied Mathematics
University of the Witwatersrand
Braamfontein, South Africa
olumuyiwa.otegbeye@wits.ac.za
https://orcid.org/0000-0003-1321-9776

Department of Mathematics
Rhodes University
Grahamstown, South Africa
s.oloniiju@ru.ac.za
https://orcid.org/0000-0002-9794-8580

(Received 19.02.2026)
(Revised 11.09.2026)
(Available online 09.11.2026)