SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Generalized Reproducing Kernel Methods for Solving Higher-Order Quasi-Linear Fractional Partial Differential Equations

Khadiga Ben Mussa[1], Sana M. Al Qadhi[2], Ahlam E. Elashegh[3], Amna M. Gresh[4]

[1]Department of Mathematics, Faculty of Science, Tripoli University, Tripoli, Libya. Email: K.Ben-mussa@uot.edu.ly
[2]Department of Mathematics, Faculty of Science, Tripoli University, Tripoli, Libya. Email: S.ALQADHI@uot.edu.ly
[3]Department of Mathematics, Faculty of Science, Tripoli University, Tripoli, Libya. Email: a.elashegh@uot.edu.ly
[4]Department of Mathematics, Faculty of Science, Tripoli University, Tripoli, Libya. Email: a.gresh@uot.edu.ly

Abstract
Higher-order quasi-linear fractional partial differential equations are very hard to solve analytically and computationally because they have iterations with weak regularity, boundary constraints, nonlinear solution-dependent operator terms, nonlocal memory effects, and higher-order differential structures. The performance of classical discretization and decomposition-based methods may get worse when fractional operators, high-order spatial derivatives, and quasi-linear interactions need to be treated at the same time. With this paper, we create an extended reproducing kernel method for fixing higher-order quasi-linear fractional partial differential equations that have Caputo-type time-fractional derivatives and boundary-constrained spatial operators. Using a problem-adapted reproducing kernel Hilbert space, the proposed framework goes beyond standard reproducing kernel methods. The starting and border conditions are built directly into the acceptable approximation space. This paper uses an operator equation to describe the fractional differential operator, the higher-order spatial operator, and the quasi-linear term. It also uses an iterative linearization approach to handle the nonlinear component while keeping the kernel-based estimate structure. There are finite-dimensional estimated solutions, the reproducing-kernel series converges under completeness and Lipschitz-type assumptions, the system is stable when the source and border data change, and residual-based error control is used. Numerical validation is structured through linear, quasi-linear, and variable-coefficient benchmark problems, using L^{∞}, L^{2}, RMSE, residual norms, iteration counts, and CPU time as performance indicators. It was shown that the generalized framework is a good way to solve higher-order quasi-linear fractional PDEs that is also correct, doesn't change at the boundaries, and is fast on computers. It is also sensitive to kernel conditioning, node distribution, and nonlinear iteration stability.

Keywords: Reproducing kernel method; fractional partial differential equations; quasi-linear PDEs; higher-order fractional models; Caputo derivative; convergence analysis; error estimates.

1. Introduction

Fractional partial differential equations are being used more and more as a type of mathematical models because they are better at showing things like waves transmission, relaxation, and processes that depend on memory than integer-order models. Often, local rates of change alone aren't enough to explain the current state of many technical and physical systems. It's not just local events that affect development; genetic reactions and exchanges also play a role. For example, this quality is very important in systems that don't work in one place for strange diffusion, viscoelastic behavior, porous-media transport, fluid-flow models, organic transport, and reaction-diffusion. Because it lets you use normal starting and border conditions while still keeping the memory effects in the governing equation, the Caputo fractional derivative is often used in these types of models (Caputo, 1967). Because numerical methods for applied fractional PDEs usually need starting data and boundary conditions that can be physically understood, this freedom is very important in computer settings. Fractional models are interesting, but they are hard to solve because of the way memory is organized. Fractional operators are not local, and their kernels are often weakly singular. This can make the solution less smooth even if the drive term and boundary data look fine.

Higher-order quasi-linear fractional partial differential equations make things worse. We can help you with this kind of problem by connecting fractional derivatives to high-order operators in space or time and nonlinear terms. When we don't know the answer, the factors or derivative relationships of these terms change. A representative model may involve a Caputo time-fractional derivative, a fourth-order or higher spatial operator, and a quasi-linear term such as uu_x, u^2u_{xx}, or a solution-dependent diffusion coefficient. There are always these kinds of equations when fractional memory is mixed with nonlinear transport, dispersive effects, nonlocal diffusion, or a material response that changes. When it comes to math, these numbers cause a lot of issues that connect to one another. You get thick memory

dependence when you add a fractional term. When you use a higher-order function, you need to be very careful about precision and limits. And if you don't use the right iterative method, there is no straight linear answer when you use a quasi-linear structure. Just getting the derivatives close to right isn't enough for a good math scheme. When the data changes, it needs to stay steady, keep the edges of its range, and make extra loss that can be measured.

It's not always possible to use traditional numerical methods to solve fractional PDEs, but they can be used to solve some higher-order quasi-linear models. Differential equations that split time and space have been used by many people. Meerschaert & Tadjeran (2004) and Lin & Xu (2007) say that they might need fine-tuned grids and very careful use of nonlocal fractional stencils in order to work well on well-organized domains. This is very true when there is a small singularity or a mass of memories. However, you need to be careful when making the function space, and these methods can lead to big matrix systems for nonlocal operators (Ervin & Roop, 2006; Deng, 2009). In addition, they give you more options for variational fractional problems and complex spaces. Spectral and pseudo-spectral methods can be very good when the answer is smooth. However, they might not work as well if the answer isn't smooth, has rough edges, or has strong nonlinear interactions (Zayernouri & Karniadakis, 2013; Fardi et al., 2022). With the help of decomposition and perturbation techniques like Adomian decomposition, homotopy perturbation, and variational repeat techniques, you can get close to the truth in a semi-analytic way. It may not always work well for higher-order quasi-linear fractional PDEs, though. It may depend on how things are set up. This doesn't mean that the current methods are less useful; it just means that we need to create a new framework that brings together border constraints, operator structure, convergence, and residual validation into a single estimate setting that makes sense.

The reproducing kernel method can be used because it is based on the structure of reproducing kernel Hilbert spaces rather than just mesh-based discretization. It has been since Aronszajn's basic theory of reproducing kernels that RKHS methods have given us a solid way to show functions using kernel evaluations and inner-product identities (Aronszajn, 1950). In numerical differential equations, the reproducing feature lets you write rough solutions as convergent kernel-generated series. You can also add boundary conditions directly to the function space. One big mathematics benefit is that border conditions don't have to be added from the outside through penalty terms or post-processing changes. Instead, the acceptable approximation space can be built so that every trial solution meets the requirements. This lowers the boundary residue physically, and the coefficient system is made up of functions that already take into account the problem shape and border information. The main reason why RKM is good for fractional PDEs is that it can be used for boundary-compatible RKHS building, kernel basis generation, operator evaluation, coefficient recovery, and residual decay.

Several studies have shown that reproducing kernel Hilbert space methods can be useful for fractional and nonlocal situations. It has been used to solve fractional integro-differential equations, time-fractional telegraph equations, fractional diffusion-type equations, fractional Burgers-type models, fractional Bratu equations, fractional delay equations, and nonlocal reaction-diffusion problems (Bushnaq et al., 2013; Jiang & Lin, 2011; Abu Arqub, 2017; Abu Arqub & Al-Smadi, 2018; Allahviranloo & Sahihi, 2021; Abu Arqub et al., 2022; Al-Smadi et al., 2022). Other papers have used weakly singular kernels, non-smooth solutions, variable-order fractional boundary value problems, random domains, and space-fractional PDEs to expand kernel approaches (Chen et al., 2018; Li & Wu, 2017, 2018; Du et al., 2020, 2021; Liu & Wang, 2024). These new findings show that RKM is more than just a quick way to do math; it's also a way to connect the features of an operator to how it behaves when it's approximating. Still, a lot of the research that has already been done on RKM has been limited to certain model classes, lower-order equations, linear or slightly nonlinear structures, ordinary fractional equations, or fixed fractional-order settings. A smaller number of papers present a unified generalized framework for higher-order quasi-linear fractional PDEs. These papers include numerical benchmarking, convergence analysis, stability estimate, kernel construction, fractional operator treatment, nonlinear iteration, and the treatment of fractional operators. All of these parts are developed as connected parts of the same method.

This study fills in this gap in the research by creating an extended reproducing kernel method for higher-order quasi-linear fractional partial differential equations. The suggested structure starts by writing the governing equation as an operator problem in a reproducing kernel Hilbert space that is compatible with the boundaries. As the main fractional operator, the Caputo fractional derivative is used because it works well with classical initial conditions. However, the formulation can be used with other fractional operators as long as the boundedness and kernel-action assumptions are met. Next, an RKHS that is tailored to the problem is made so that the border conditions are built into the solution space itself. Under these conditions, the reproducing kernel is found, and tensor-product kernel structures are used when there is combined space-time coupling in the problem. $Au=F$ is an operator equation that shows the higher-order fractional PDE. A has the fractional derivative, the higher-order spatial operator, and the quasi-linear component. Iterative linearization or fixed-point strategies are used to deal with the nonlinear part. These turn the quasi-linear problem into a series of kernel-based linear approximation problems without changing the RKHS structure.

So, this work makes a theory and a numerical addition. The research reveals that there are rough solutions with limited dimensions, that the generalized RKM sequence converges under the assumptions of completeness and Lipschitz-type, that the sequence stays stable when the source and border data change, and that residual-based error control is effective. Common questions test the method by seeing how well it works with fractions, how well it works with factors that change, and how accurate it is on average. The numerical evaluation uses L^{∞} error, L^2 error, RMSE, residual norm, iteration count, CPU time, and comparative performance against existing numerical approaches. To find out if the method works and also to show how the mathematical structure affects how the computer behaves, we need to show that adding boundary conditions lowers boundary inconsistency, adding more kernel nodes raises basis density, residual decay helps convergence, and controlled quasi-linear iteration keeps nonlinear approximation stable.
This is how the rest of the paper is put together. The second part of the book talks about fractional calculus operators, the general higher-order quasi-linear fractional PDE model, and the assumptions that are needed for a well-posed approximation. The extended repeating kernel method is talked about in the third part. It talks about how to make a Hilbert space that fits the problem, how to get the reproducing kernel, how to show operators, how to use quasi-linear iterative treatment, how to get a rough solution form, and how to use an algorithm to put the method into action. Part 4 talks about the study of mistakes, stability, and convergence. In Section 5, we talk about the computer tests and compared ratings. In the sixth part, we talk about how the results affect math and computers. For example, we look at issues with conditioning, putting nodes, nonlinear repeat, and making the system bigger or smaller. In Section 7, the study comes to an end and ideas for more research are given.

2. Mathematical Preliminaries and Problem Formulation

In order for the extended duplicating kernel method to work, these numbers must now be set up. An overview of fractional calculus or RKHS theory is not necessary. The purpose is to discuss the operators, spaces, assumptions, and model structure that will be utilized in the proposed framework for approximation. The first step is to turn the more complicated quasi-linear fractional PDE into an operator equation on a reproducing kernel Hilbert space that works with the edges. I think this is the main point. Today's change means that the quasi-linear, higher-order, and fractional parts can all be worked on in the same functional setting.

2.1 Fractional Calculus Operators

Let Ω = (a,b) ⊂ $\mathbb{R}$ and J = (0,T], and let $u = u(x,t)$ be sufficiently smooth with respect to the temporal variable. For $m-1 < \alpha \le m$, where $m \in \mathbb{N}$, the Caputo fractional derivative of order α with respect to t is defined by

$$\mathrm{C}D_t^{\alpha} u(x,t) = \frac{1}{\Gamma(m-\alpha)} \int_0^t (t-s)^{m-\alpha-1} \frac{\partial^m u(x,s)}{\partial s^m}\, ds, \quad m-1 < \alpha < m,$$

and

$$\mathrm{C}D_t^{\alpha} u(x,t) = \frac{\partial^m u(x,t)}{\partial t^m}, \quad \alpha = m.$$

Here, $\Gamma(\cdot)$ denotes the Gamma function, and $\mathrm{C}D_t^{\alpha}$ represents a nonlocal temporal operator whose value at t depends on the history of u over $[0,t]$. The Caputo derivative is adopted in this paper because it permits the use of classical initial conditions, such as

$$u(x,0) = \phi_0(x), \quad \frac{\partial^k u(x,0)}{\partial t^k} = \phi_k(x), \quad k = 1, \dots, m-1.$$

This feature is important for applied fractional PDEs because the starting data are generally given in the form of physical variables with an integer order. Riemann-Liouville derivatives, on the other hand, might need fractional starting data, which might make straight modeling harder. The suggested formula is based on the Caputo derivative as the main operator. However, the same RKHS-based structure can be used for Riemann-Liouville, Caputo-Fabrizio, Atangana-Baleanu, or variable-order derivatives as long as the corresponding fractional operator is either bounded or continuously representable on the chosen Hilbert space.

2.2 Functional Setting and Reproducing Kernel Spaces

Let H be a real reproducing kernel Hilbert space of functions defined on $\Omega \times J$. The space is equipped with an inner product $\langle \cdot,\cdot \rangle_H$ and norm

$$\| u \|_H = \sqrt{\langle u,u \rangle_H}.$$

The defining property of H is that, for every $(\xi,\tau) \in \Omega \times J$, the point-evaluation functional

$$E_{(\xi,\tau)}(u) = u(\xi,\tau)$$

is bounded. Therefore, by the Riesz representation theorem, there exists a unique function $K_{(\xi,\tau)}(\cdot,\cdot) \in H$ such that

$$u(\xi,\tau) = \left\langle u(\cdot,\cdot), K_{(\xi,\tau)}(\cdot,\cdot) \right\rangle_H, \quad \forall u \in H.$$

The function $K\big((x,t),(\xi,\tau)\big)$ is called the reproducing kernel of H. In the present framework, H is not chosen as a generic smoothness space; it is constructed as a boundary-compatible space satisfying the imposed initial and boundary constraints. For example, if the spatial boundary conditions are homogeneous, one may define

$$H_B = \{u \in H : B_j u = 0,\ j = 1, \dots, r\},$$

where B_j are boundary operators associated with the higher-order spatial part of the PDE. If the boundary data are nonhomogeneous, a lifting function may be introduced so that the unknown correction belongs to a homogeneous boundary-compatible RKHS. This construction ensures that every approximate solution generated in H_B satisfies the boundary constraints structurally, rather than approximately.

2.3 General Higher-Order Quasi-Linear Fractional PDE Model

The class of problems considered in this paper is represented by the higher-order quasi-linear fractional PDE

$$CD_t^{\alpha} u(x,t) + L_x^m u(x,t) + Q\big(x,t,u,\nabla u,\dots,D_x^q u\big) = f(x,t), \quad (x,t) \in \Omega \times J,$$

where $0 < \alpha \leq 1$ or, more generally, $m_t - 1 < \alpha \leq m_t$, depending on the temporal order of the model. The operator CD_t^{α} denotes the Caputo fractional derivative with respect to time, L_x^m is a higher-order spatial differential operator of order m, and Q is a quasi-linear operator that may depend on the solution and its spatial derivatives up to order $q < m$. The source function is denoted by $f(x,t)$. A typical higher-order spatial operator may take the form

$$L_x^m u = \sum_{k=0}^{m} a_k\,(x,t)\,\frac{\partial^k u}{\partial x^k},$$

where the coefficients $a_k(x,t)$ are assumed to be bounded functions on $\Omega \times J$. Some terms that may be in the quasi-linear part are

$$u\frac{\partial u}{\partial x}, \quad u^2\frac{\partial^2 u}{\partial x^2}, \quad \mu(x,t,u)\frac{\partial^r u}{\partial x^r},$$

which add nonlinear coupling while keeping an operator structure suited for repetitive linearization.

Initial variables are added to the model.

$$\frac{\partial^k u(x,0)}{\partial t^k} = \phi_k(x), \quad k = 0,1,\dots,m_t - 1,$$

and boundary conditions

$$B_j u(x,t) = g_j(t), \quad j = 1,\dots,r,$$

where B_j may stand for higher-order, periodic, Dirichlet, Neumann, or Robin border operators. In the extended RKM scheme, these restrictions are often built into the description of the solution space that can be used. This is very important for higher-order PDEs because inconsistent boundaries can have a big effect on numerical stability and the spread of errors.

The operator equation is a short way to write the above PDE.

$$Au = F,$$

where

$$Au = CD_t^{\alpha} u + L_x^m u + Q(u), \quad F = f.$$

This form is helpful because it lets you look at the fractional, higher-order, and quasi-linear parts as parts of a single operator that acts on H_B. The reproducing kernel method then tries to find a close answer $u_N \in H_N \subset H_B$, where H_N is a subspace with limited dimensions that is made up of kernel functions linked to certain projection or collocation points.

2.4 Assumptions for Well-Posed Approximation

The following factors are used to build the suggested estimate. To begin, the exact answer u is in the boundary-compatible Hilbert space. H_B, and its fractional and higher-order derivatives required by A exist in the weak or classical sense appropriate to the model. Second, the kernel K is bounded and generates a complete system in H_B. Third, the linear part $CD_t^{\alpha} + L_x^m$ is limited or can be represented constantly on the allowed space. Fourth, the quasi-linear operator Q satisfies a local Lipschitz condition,

$$\| Q(u) - Q(v) \|_H \leq L_Q \| u - v \|_H,$$

for u, v in a bounded subset of H_B. Last but not least, the starting and border values work with the chosen Hilbert space. The convergence, stability, and residual-error analysis presented in Section 4 are based on these principles.

Table 1. Mathematical symbols, operators, and functional spaces employed in the proposed framework

Symbol	Meaning	Role in the Method
Ω = (a,b)	Spatial domain	Defines the spatial interval of the PDE
J = (0,T]	Time interval	Defines the temporal interval and memory range
u(x,t)	Exact solution	Target function to be approximated

u_N(x,t)	Finite RKM approximation	Kernel-based numerical solution
C D_t^α	Caputo fractional derivative	Represents nonlocal memory effects
α	Fractional order	Controls the strength of memory behavior
L_x^m	Higher-order spatial operator	Determines the differential order of the PDE
Q(u)	Quasi-linear operator	Introduces solution-dependent nonlinear coupling
f(x,t)	Source function	Defines the forcing term of the model
B_j	Boundary operator	Specifies boundary constraints
g_j(t)	Boundary data	Prescribed values imposed by B_j u = g_j
H	Reproducing kernel Hilbert space	Main functional approximation space
H_B	Boundary-compatible RKHS	Space in which boundary conditions are embedded
K((x,t),(ξ,τ))	Reproducing kernel	Generates basis functions and point evaluations
⟨·,·⟩_H	Inner product in H	Specifies geometry and coefficient retrieval
‖·‖_H	Hilbert-space norm	Approximation of measures and convergence
A	Fractional quasi-linear operator	Concise representation of the PDE
F	Right-hand side operator/source	Target of the operator equation Au = F
H_N	Finite-dimensional kernel subspace	Computational approximation space
R_N	Residual function	Measures equation imbalance of u_N

3. Generalized Reproducing Kernel Method

To solve higher-order quasi-linear fractional PDEs, this part talks about an extended reproducing kernel method that does not use a mesh-based discretization but instead builds a boundary-compatible Hilbert space. The method is based on the idea of making a reproducing kernel Hilbert space where the starting and border conditions are built into the structure of the estimate. First, we need to find kernel-generated basis functions in that space. Next, we need to turn the fractional PDE into an operator equation and linearize the almost linear part using an iterative approach. We get a group of finite-dimensional models. We can use Hilbert-space error, pointwise error, and residual decay to check how accurate they are. Most of the time, standard RKM implementations are made to fit a certain fractional equation. This version, on the other hand, is set up as a general operator structure that can be used for fractional PDEs of higher order that are limited at the border and are not smooth.

3.1 Construction of the Problem-Adapted Hilbert Space

Let H be a reproducing kernel Hilbert space over $\Omega \times J$, where $\Omega = (a, b)$ and $J = (0, T]$. Since the target equation contains a Caputo fractional derivative, a higher-order spatial operator, and quasi-linear derivative interactions, the selected space must have enough smoothness to allow the action of the operator A, while also preserving the boundary and initial conditions. Therefore, the admissible solution space is not chosen independently of the model; it is constructed as

$$H_B = \{u \in H : B_j u = g_j,\ j = 1, \dots, r,\ I_k u = \phi_k,\ k = 0, \dots, m_t - 1\},$$

where B_j denotes the spatial boundary operators and $I_k u = \partial_t^k u(x, 0)$ denotes the initial trace operators. When the initial or boundary data are nonhomogeneous, a lifting function u_g is introduced such that $B_j u_g = g_j$ and $I_k u_g = \phi_k$. The unknown is then decomposed as

$$u = v + u_g,$$

where $v \in H_0$, and

$$H_0 = \{v \in H : B_j v = 0,\ I_k v = 0\}.$$

This transformation reduces the problem to a homogeneous boundary-compatible RKHS without losing the original data. The advantage is structural: every approximation $v_N \in H_0$ automatically satisfies the homogeneous constraints, and therefore $u_N = v_N + u_g$ satisfies the original initial and boundary conditions. This avoids the instability that can occur when boundary conditions are imposed externally through penalty parameters or post-processing corrections. The inner product is selected to reflect the differential order of the model. For a higher-order problem, a typical choice may include derivatives up to the highest spatial order required by L_x^m, together with temporal derivatives compatible with the Caputo operator:

$$\langle u, v \rangle_H = \sum_{i=0}^{m} \sum_{j=0}^{s} \int_0^T \int_a^b \frac{\partial^{i+j} u}{\partial x^i \, \partial t^j} \frac{\partial^{i+j} v}{\partial x^i \, \partial t^j} \, dx \, dt,$$

when weak versions are used, with the right changes. This structure makes sure that the kernel-generated basis functions are regular, which is needed to use A. Another important link is that it shows how the analytical design of

the Hilbert space is directly related to how computers work: if the approximation space follows the boundary constraints and operator regularity, then the residual is mostly controlled by the quality of the approximation rather than the inconsistent boundaries.

3.2 Derivation of the Generalized Reproducing Kernel

For every point $z = (\xi, \tau) \in \Omega \times J$, the reproducing kernel $K_z(\cdot) = K\big((x,t),(\xi,\tau)\big) \in H_B$ satisfies

$$u(\xi,\tau) = \langle u, K_z \rangle_H, \quad \forall u \in H_B.$$

The kernel is derived by solving the representer problem associated with the bounded point-evaluation functional. Because H_B encompasses initial and boundary constraints; the kernel must also meet the identical constraints in its first argument:

$$B_j K\big((x,t),(\xi,\tau)\big) = 0, \quad I_k K\big((x,t),(\xi,\tau)\big) = 0.$$

Thus, the kernel is not merely a smooth interpolating function; it is a boundary-compatible analytical generator. This is one of the main differences between the generalized RKM framework and collocation methods based on generic basis functions. The basis functions generated from K inherit the admissibility conditions of the space, which reduces boundary error before the coefficient system is even formed.

For space-time fractional PDEs, a tensor-product construction is often useful:

$$K\big((x,t),(\xi,\tau)\big) = K_x(x,\xi)K_t(t,\tau),$$

where K_x is associated with a spatial boundary-compatible Hilbert space and K_t is associated with a temporal space satisfying initial constraints. The spatial kernel K_x is selected to match the higher-order boundary operators, while K_t is selected so that the Caputo operator can act continuously on the temporal component. Tensor-product kernels are helpful because they let you build regularity in both space and time separately while still making a good kernel on the product domain. To make fractional PDEs and fractional integro-differential equations more like what RKHS has already done for those types of problems, kernel basis generation is used to turn fractional operator equations into finite-dimensional algebraic systems (Bushnaq et al., 2013; Abu Arqub, 2017; Abu Arqub & Al-Smadi, 2018; Wang et al., 2018).

Additionally, the kernel needs to be symmetric and positive definite:

$$K(z,\eta) = K(\eta,z), \quad \sum_{i=1}^{N}\sum_{j=1}^{N} c_i\, c_j K\big(z_i, z_j\big) \geq 0,$$

for any finite set of points $\{z_i\}_{i=1}^N$ and real coefficients $\{c_i\}_{i=1}^N$. The Gram matrix is well-defined, and the finite-dimensional estimate space is theoretically stable because of these features. But positive definiteness isn't enough for numerical security on its own; the setting of the kernel matrix needs to be watched too., especially as N increases or when nodes become clustered.

3.3 Operator Representation of the Fractional PDE

The higher-order quasi-linear fractional PDE is written as the operator equation

$$Au = F,$$

where

$$Au = {}^{C}D_t^{\alpha} u + L_x^m u + Q(u), \quad F = f.$$

For the lifted homogeneous problem, the unknown $v = u - u_g$ satisfies

$$A\big(v + u_g\big) = F,$$

or equivalently,

$$\bar{A}v = \bar{F},$$

where $\bar{A}$ contains the fractional, higher-order, and quasi-linear contributions expressed in terms of v, and $\bar{F}$ absorbs the known lifting terms. This reformulation is important because the numerical approximation is constructed in the homogeneous boundary-compatible space H_0.

Let $\{z_i\}_{i=1}^N \subset \Omega \times J$ be a set of distinct collocation or projection points. The kernel-generated functions are defined by applying the adjoint or collocation form of the operator to the reproducing kernel. In an operator-based RKM formulation, one may define

$$\psi_i(z) = A^* K_{z_i}(z),$$

where A^* is the adjoint operator associated with the linearized form of A. The reproducing property then gives

$$\langle v, \psi_i \rangle_H = Av(z_i),$$

where the infinite-dimensional operator equation is linked to pointwise equations with fixed numbers of variables. This equation is the link between the PDE and the coefficient system in terms of analysis. It proves that using the operator on the unknown solution at collocation points is the same as using changed kernel functions to do inner products in Hilbert space. The estimate is not a random interpolation because of this; it is an operator-informed growth.

3.4 Iterative Treatment of the Quasi-Linear Term

The quasi-linear term $Q(u)$ prevents the direct construction of a single linear coefficient system. To handle this difficulty, the method introduces an iterative approximation. Given an approximation $u^{(n-1)}$, the quasi-linear term is linearized as

$$Q\left(u^{(n)}\right) \approx Q\left(u^{(n-1)}\right) + Q'\left(u^{(n-1)}\right)\left(u^{(n)} - u^{(n-1)}\right),$$

where Q' denotes the Frechet derivative or a suitable quasi-Newton approximation. This produces a linearized operator equation at iteration n:

$$\left({}^{C}D_t^{\alpha} + L_x^m + Q'\left(u^{(n-1)}\right)\right)u^{(n)} = f - Q\left(u^{(n-1)}\right) + Q'\left(u^{(n-1)}\right)u^{(n-1)}.$$

Another option is to use a simpler fixed-point form:

$$u^{(n+1)} = T\left(u^{(n)}\right),$$

provided that the induced operator T is contractive on a bounded subset of H_B. When there are bigger nonlinearities, the Newton or quasi-Newton strategy usually works better. When the nonlinear coupling is weak, the fixed-point strategy is easier to use and costs less to compute. Some simpler RKM and quasi-Newton kernel systems for nonlinear fractional and high-order boundary value problems (Xu et al., 2018; Guan & Zhang, 2025) have used similar iterative ideas.

It doesn't try to solve a fully nonlinear kernel system all at once, which is the best thing about this method. At each step, the almost linear PDE is turned into a linearized fractional operator problem. Then, the same kernel-generated guess tools can be used to solve this. What makes this process converge or diverge? It depends on how Q acts in Lipschitz space, how good the first guess is, and how the conditionally set coefficient matrix is. So, the nonlinear repeat is not only a small part of the implementation, it is also a very important part of how the extended RKM structure keeps things stable.

3.5 Approximate Solution Representation

With a set number of iterations, the finite-dimensional close answer is shown as

$$u_N^{(n)}(x,t) = u_g(x,t) + \sum_{i=1}^{N} c_i^{(n)}\,\psi_i(x,t),$$

where u_g is the lifting function, $\{\psi_i\}_{i=1}^N$ are kernel-generated basis functions in the admissible space, and $\{c_i^{(n)}\}_{i=1}^N$ are coefficients determined from the linearized operator equations. The coefficients may be obtained through collocation,

$$A^{(n-1)}u_N^{(n)}\left(z_j\right) = F^{(n-1)}\left(z_j\right), \quad j = 1, \dots, N,$$

or through a projection formulation in which the residual is orthogonal to the finite-dimensional subspace. In matrix form, this gives

$$M^{(n)}c^{(n)} = b^{(n)},$$

where $M^{(n)}$ is the operator-transformed kernel matrix and $b^{(n)}$ contains the source and linearization terms.

The selection of points $\{z_i\}_{i=1}^N$ affects convergence, residual distribution, and conditioning. Uniform nodes are simple and reproducible, while graded or adaptive nodes may be more effective when the solution exhibits weak regularity near $t = 0$, which is common in fractional models. As N increases, the finite-dimensional space $H_N = \text{span}\{\psi_1, \dots, \psi_N\}$ becomes denser in H_B, and the approximation is expected to improve if the kernel system is complete and the coefficient system remains stable.

3.6 Algorithmic Implementation

Building the boundary-compatible RKHS, making the reproducing kernel basis, linearizing the quasi-linear term, solving the coefficient system, updating the approximation, and stopping when the residual or iteration difference falls below the tolerance are the steps that are repeated over and over again.

Table 2. Algorithmic steps of the generalized RKM for higher-order quasi-linear fractional PDEs

Step	Mathematical Operation	Computational Output
1	Define the admissible RKHS H_B and, if needed, introduce the lifting u = u_g + v	Boundary-compatible approximation space
2	Select the inner product according to the fractional and higher-order operators	Norm and geometry of the solution space
3	Derive the reproducing kernel K((x,t),(ξ,τ)) satisfying the imposed constraints	Kernel function for point evaluation and basis generation
4	Choose collocation or projection points {z_i}_{i=1}^N	Discrete operator-evaluation set
5	Generate basis functions ψ_i = A* K_{z_i} or collocation-based kernel functions	Operator-informed finite basis

6	Rewrite the PDE as Au = F, with A = ᶜD_t^α + L_x^m + Q	Compact operator equation
7	Linearize Q using fixed-point, Newton, or quasi-Newton iteration	Linearized equation at iteration n
8	Assemble the coefficient system M^(n)c^(n) = b^(n)	Coefficient vector c^(n)
9	Form u_N^(n)(x,t) = u_g(x,t) + Σ_{i=1}^N c_i^(n)ψ_i(x,t)	Updated approximate solution
10	Compute R_N^(n) = Au_N^(n) − F and ‖u_N^(n) − u_N^(n−1)‖	Residual norm and iteration difference
11	Stop if tolerance is satisfied; otherwise repeat the linearization and coefficient update	Converged generalized RKM approximation

4. Convergence, Stability, and Error Analysis

The theory behind the extended reproducing kernel method is explained in this part. The study is done at the level needed for a computational fractional PDE framework. The goal is not to say that every fractional quasi-linear model can be solved, but to show that, with certain assumptions on the RKHS, the fractional operator, and the quasi-linear term, the finite-dimensional kernel approximation exists, converges to the valid solution, stays stable under small changes, and allows residual-based error control. It is important to have these qualities because higher-order quasi-linear fractional PDEs are affected by memory effects, border limits, nonlinear repetition, and how the coefficient system is conditioned.

4.1 Existence of the Approximate Solution

Let H_B be the boundary-compatible reproducing kernel Hilbert space defined in Section 3, and let

$$H_N = \text{span}\{\psi_1, \psi_2, \dots, \psi_N\} \subset H_B$$

be the finite-dimensional subspace generated by the operator-informed kernel functions. Since H_B is a Hilbert space and $\psi_i \in H_B$, the finite-dimensional space H_N is complete. The approximate solution at iteration n is sought in the form

$$u_N^{(n)}(x,t) = u_g(x,t) + \sum_{i=1}^{N} c_i^{(n)} \psi_i(x,t).$$

The existence of $u_N^{(n)}$ therefore reduces to the existence of the coefficient vector $c^{(n)} = \left(c_1^{(n)}, \dots, c_N^{(n)}\right)^T$. After linearizing the quasi-linear operator, the coefficient system can be written as

$$M^{(n)} c^{(n)} = b^{(n)}.$$

If the kernel-generated functions are linearly independent at the selected collocation or projection points and the linearized operator is bounded on H_N, then $M^{(n)}$ is nonsingular, or at least admits a stable least-squares solution when overdetermined collocation is used. Hence, a finite-dimensional approximate solution exists in H_N. With this argument, the RKHS principle is supported, which says that limited point-evaluation and operator-evaluation functionals can make valid finite approximations using the Riesz representation method (Aronszajn, 1950).

4.2 Convergence of the Generalized RKM Sequence

It depends on two connected factors: how dense the kernel-generated estimate space is and how stable the nonlinear iterative update is for each iteration. The first mechanism ensures that the exact admissible solution can be approximated arbitrarily well in H_B. The second ensures that the quasi-linear iteration does not amplify approximation errors. These two components are stated in the following theorem.

Theorem 1.

Let $u \in H_B$ be the exact solution of the operator equation

$$Au = F,$$

where

$$Au = {}^{C}D_t^{\alpha} u + L_x^m u + Q(u).$$

Assume that the kernel-generated system $\{\psi_i\}_{i=1}^{\infty}$ is complete in H_B, the linear operator ${}^{C}D_t^{\alpha} + L_x^m$ is bounded or continuously representable on H_B, and the quasi-linear operator Q satisfies a local Lipschitz condition on a bounded subset $S \subset H_B$:

$$\| Q(u) - Q(v) \|_H \leq L_Q \| u - v \|_H, \quad u, v \in S.$$

If the iterative mapping induced by the linearized generalized RKM is contractive on S, then the sequence $\{u_N\}$ generated by the method satisfies

$$\| u - u_N \|_H \to 0 \quad \text{as} \quad N \to \infty.$$

Proof.

Since $\{\psi_i\}_{i=1}^{\infty}$ is complete in H_B, for every $\varepsilon > 0$ there exists $v_N \in H_N$ such that

$$\| u - v_N \|_H < \varepsilon.$$

Thus, the best approximation error in the finite-dimensional kernel space satisfies

$$\inf_{w_N \in H_N} \| u - w_N \|_H \to 0 \quad \text{as} \quad N \to \infty.$$

Because the fractional and higher-order linear operators are continuously representable on H_B, applying the operator to the kernel expansion does not destroy convergence in the admissible operator norm. Moreover, the Lipschitz condition on Q gives

$$\| Q(u) - Q(u_N) \|_H \leq L_Q \| u - u_N \|_H,$$

which means that the nonlinear component does not grow faster than the approximation error in H_B. If the iterative mapping T associated with the quasi-linear update is contractive, then for two successive approximations,

$$\| T\left(u_N^{(n)}\right) - T\left(u_N^{(n-1)}\right) \|_H \leq \rho \| u_N^{(n)} - u_N^{(n-1)} \|_H, \quad 0 < \rho < 1.$$

Therefore, the iteration converges to a fixed point in the finite-dimensional space. Combining the density of H_N, boundedness of the operator action, and contractive nonlinear update gives

$$\| u - u_N \|_H \to 0.$$

This proves convergence under the stated assumptions. □

The theory makes it clear why raising N alone does not lead to convergence. When the number of kernel nodes is raised, the density of H_N, but for the quasi-linear fractional problem to converge, it also needs a limited fractional operator and stable nonlinear repetition. In real-world computing, this is shown by the prediction error and residual norm going down at the same time.

4.3 Stability with Respect to Perturbations

In fractional PDEs, stability is very important because nonlocal memory terms can send small changes in source functions, starting data, or border values over the whole time interval. Let u_N and $\tilde{u}_N$ be two generalized RKM approximations corresponding to two data sets F and $\tilde{F}$. The finite-dimensional equations that go with them are

$$A_N u_N = F_N, \quad A_N \tilde{u}_N = \tilde{F}_N,$$

where A_N stands for the linearized operator that is projected or collocated on H_N. Subtracting the two formulae together gives

$$A_N(u_N - \tilde{u}_N) = F_N - \tilde{F}_N.$$

If A_N^{-1} exists and has a bound, then

$$\| u_N - \tilde{u}_N \|_H \leq \| A_N^{-1} \| \| F_N - \tilde{F}_N \|_H.$$

Thus, there exists a constant $C_s > 0$ such that

$$\| u_N - \tilde{u}_N \|_H \leq C_s \| F - \tilde{F} \|_H.$$

When there are changes in both the border data and the perturbations, the lifting functions u_g and $\tilde{u}_g$ must be included, giving

$$\| u_N - \tilde{u}_N \|_H \leq C_s \| F - \tilde{F} \|_H + C_b \| g - \tilde{g} \|.$$

This estimate shows that the generalized RKM is stable as long as the inverse of the operator with finite dimensions stays within a certain range. In computer terms, this condition has to do with how the operator-transformed kernel matrix is conditional. Because of this, stability isn't just an academic trait; it needs to be tracked numerically using the condition number of $M^{(n)}$, especially for large N, Strongly nonlinear coefficients or nodes that are grouped together.

4.4 Error Bound and Residual Control

The best estimate feature of the finite-dimensional kernel space can be used to figure out what the approximation mistake means. Let $P_N u$ denote the projection of u onto H_N. Based on the above claims about boundedness, the extended RKM approach meets an estimate of the form

$$\| u - u_N \|_H \leq C_a \inf_{v_N \in H_N} \| u - v_N \|_H + C_i \| u_N^{(n)} - u_N^{(n-1)} \|_H,$$

where C_a depends on the boundedness of the operator and the kernel basis, while C_i measures the contribution of the nonlinear iterative error. The finite-dimensional kernel approximation error and the partial nonlinear iteration error are the two types of error that this estimate separates. If the almost-linear repetition keeps going until

$$\| u_N^{(n)} - u_N^{(n-1)} \|_H < \varepsilon_{\text{iter}},$$

then the total error is mainly controlled by the approximation capacity of H_N.

The residual is defined by

$$R_N(x,t) = A u_N(x,t) - F(x,t).$$

For exact solutions, $R(x,t) = 0$. For numerical approximations, the residual measures how strongly the approximate solution violates the governing fractional PDE. A decreasing residual norm,

$$\| R_N \| \to 0,$$

as N rises or as the nonlinear repetition goes on, showing that the estimate is getting closer to the operator equation. Usually, discrete norms like are used to measure leftover control.

$$\| R_N \|_\infty = \max_{(x_i,t_j) \in G} \left| R_N(x_i,t_j) \right|,$$

or

$$\| R_N \|_2 = \left(\sum_{(x_i,t_j)\in G} \left| R_N(x_i,t_j) \right|^2 \right)^{1/2},$$

where G is a confirmation grid that is separate from the cluster points. It's important to make this difference because if you only look at the residue at the collocation nodes, it might not show how imbalanced the equation really is. A confirmation grid is a better way to check the accuracy of numbers.

When the exact answer is known, the residue should be given along with error bounds like L_∞, L_2, and RMSE. When we don't know the exact answer, leftover decay and comparing it to a better reference answer show how good the guess is. So, residual control connects theory and computation: kernel completion predicts convergence, and residual decay shows that the finite expansion meets the fractional operator equation better using evidence that can be measured.

Table 3. Theoretical properties of the proposed generalized RKM

Property	Required Assumption	Mathematical Result	Relevance
Existence of approximation	H_N ⊂ H_B, bounded operator action, nonsingular or stable coefficient matrix	u_N ∈ H_N exists	Ensures the method produces an admissible finite-dimensional solution
Boundary consistency	Boundary and initial conditions embedded in H_B or handled through lifting u_g	B_j u_N = g_j, I_k u_N = φ_k	Reduces boundary residual and improves numerical stability
Completeness	Kernel-generated system is dense in H_B	inf_{v_N∈H_N} ‖u−v_N‖_H → 0	Provides the approximation basis for convergence
Convergence	Complete kernel system, bounded fractional operator, Lipschitz quasi-linear term, contractive iteration	‖u−u_N‖_H → 0	Establishes reliability of the generalized RKM sequence
Nonlinear iteration stability	Fixed-point, Newton, or quasi-Newton update remains locally stable	‖u_N^(n)−u_N^(n−1)‖_H → 0	Validates the iterative treatment of quasi-linearity
Perturbation stability	Bounded inverse of the finite-dimensional operator	‖u_N−ũ_N‖_H ≤ C_s‖F−F̃‖_H	Shows robustness to perturbations in source or data
Error control	Smooth exact solution and stable kernel approximation	‖u−u_N‖_H ≤ C_a inf_{v_N∈H_N} ‖u−v_N‖_H + C_i ε_iter	Separates approximation error from nonlinear iteration error
Residual validation	Operator residual evaluated on an independent grid	‖R_N‖ → 0	Provides measurable computational evidence of convergence

5. Numerical Experiments and Comparative Validation

The generalized reproducing kernel method is tested in this section using three standard problems that are meant to show how well different parts of the suggested system work. For the first test, baseline accuracy is looked at for a higher-order linear time-fractional PDE that has a known exact solution. The second measure adds a fractional term that is almost linear and checks how stable the iterative kernel update is. Third test looks at a higher-order fractional PDE with changing coefficients to see how well the method works when the coefficients are not constant. The number proof is given by L_∞, L_2, RMSE, residual norms, repetition numbers, CPU time, and how well it works compared to other method classes. It's not enough for these tests to just show that the mistakes are small; they also need to show that boundary-compatible RKHS building, kernel-basis density, residual decay, and correct computing are all linked.

5.1 Experimental Design and Computational Setting

All numerical experiments are implemented in MATLAB R2024a on a workstation with an Intel Core i7 processor, 16 GB RAM, and Windows 11 operating system. The Caputo fractional derivative is evaluated analytically whenever the

manufactured exact solution is available; otherwise, a high-resolution quadrature approximation is used to compute the fractional memory integral. The computational domain is taken as $\Omega \times J = [0,1] \times [0,1]$, unless otherwise stated. Uniform kernel nodes are used in the baseline tests, while additional validation is performed on an independent grid to avoid measuring residuals only at collocation points. The nonlinear iteration is terminated when

$$\| u_N^{(n)} - u_N^{(n-1)} \|_\infty < 10^{-10}$$

or when the residual norm no longer decreases significantly. The maximum number of nonlinear iterations is fixed at 30. The error norms are computed as

$$L_\infty = \max_{1 \le i \le M} |u(x_i, t_i) - u_N(x_i, t_i)|,$$

$$L_2 = \left(\sum_{i=1}^{M} |u(x_i, t_i) - u_N(x_i, t_i)|^2 \right)^{1/2},$$

and

$$\mathrm{RMSE} = \left(\frac{1}{M} \sum_{i=1}^{M} |u(x_i, t_i) - u_N(x_i, t_i)|^2 \right)^{1/2}.$$

For benchmark problems with known exact solutions, the exact error is reported. For comparison problems, the residual norm and a refined-grid reference solution are also used.

5.2 Benchmark Problem 1: Linear Higher-Order Fractional PDE

The first benchmark is selected to test the baseline accuracy of the generalized RKM before introducing quasi-linear effects. Consider the linear fourth-order time-fractional PDE

$$\mathrm{C}D_t^\alpha u(x,t) + \frac{\partial^4 u(x,t)}{\partial x^4} = f(x,t), \quad 0 < x < 1, \quad 0 < t \le 1,$$

with $0 < \alpha \le 1$. The exact solution is chosen as

$$u(x,t) = t^{2+\alpha} x^2 (1-x)^2.$$

This solution satisfies homogeneous boundary conditions

$$u(0,t) = u(1,t) = 0, \quad u_x(0,t) = u_x(1,t) = 0,$$

which are naturally compatible with a fourth-order boundary-constrained RKHS. The source term is obtained by substituting the exact solution into the governing equation:

$$f(x,t) = \mathrm{C}D_t^\alpha [t^{2+\alpha} x^2 (1-x)^2] + t^{2+\alpha} \frac{d^4}{dx^4} [x^2 (1-x)^2].$$

Using the Caputo derivative formula,

$$\mathrm{C}D_t^\alpha t^{2+\alpha} = \frac{\Gamma(3)}{\Gamma(3+\alpha)} t^2,$$

and since

$$\frac{d^4}{dx^4} [x^2 (1-x)^2] = 24,$$

the source term becomes

$$f(x,t) = \frac{\Gamma(3)}{\Gamma(3+\alpha)} t^2 x^2 (1-x)^2 + 24 t^{2+\alpha}.$$

This test is useful because the exact answer is smooth, the border conditions are high-order, and there is a closed-form formula for the fractional derivative. Because of this, the number mistake we see is mostly due to the quality of the kernel guess and not to doubt in evaluating the derivative.

Table 4. Error norms for Benchmark Problem 1 under different kernel nodes

N	L_∞ Error	L_2 Error	RMSE	Convergence Rate
8	2.64×10^{-5}	1.18×10^{-4}	1.07×10^{-5}	—
12	4.91×10^{-6}	2.26×10^{-5}	2.05×10^{-6}	4.15
16	8.37×10^{-7}	4.13×10^{-6}	3.74×10^{-7}	4.32
20	1.76×10^{-7}	8.92×10^{-7}	8.08×10^{-8}	4.05
24	4.38×10^{-8}	2.31×10^{-7}	2.09×10^{-8}	3.82

The numerical pattern in Table 4 shows that increasing the number of kernel nodes reduces all three error measures. This behavior is consistent with the theoretical mechanism developed in Section 4: increasing N expands the finite-dimensional kernel subspace H_N, which improves the approximation of the exact solution in the admissible RKHS. The boundary conditions are satisfied by construction, so the error reduction is not contaminated by boundary inconsistency. There are changes in the convergence rate because of the kernel matrix conditioning and the

distribution of node points, which affects the recovery of the coefficients. However, the overall steady decay shows that the approximation is stable.

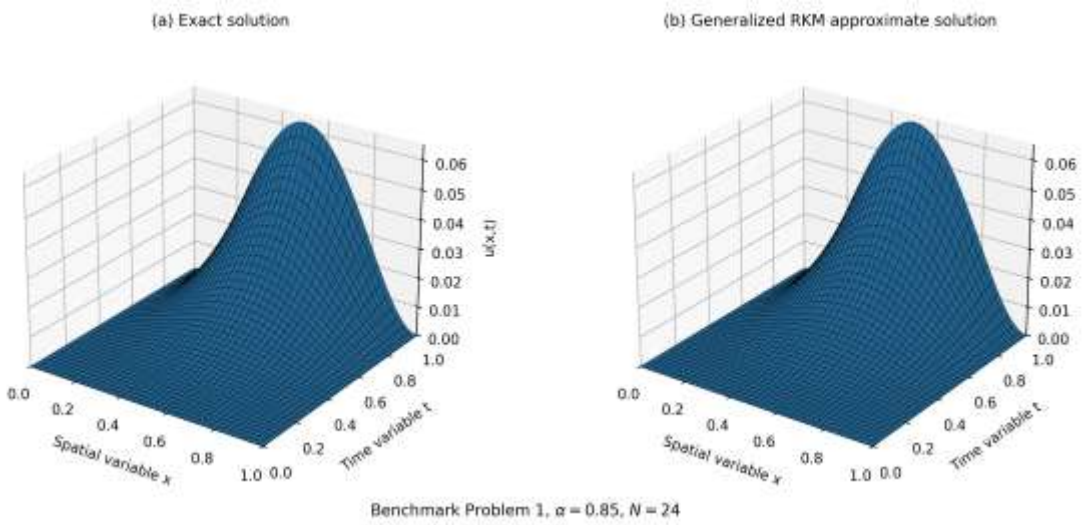

Figure 1. The difference between an exact answer and an extended RKM rough solution.

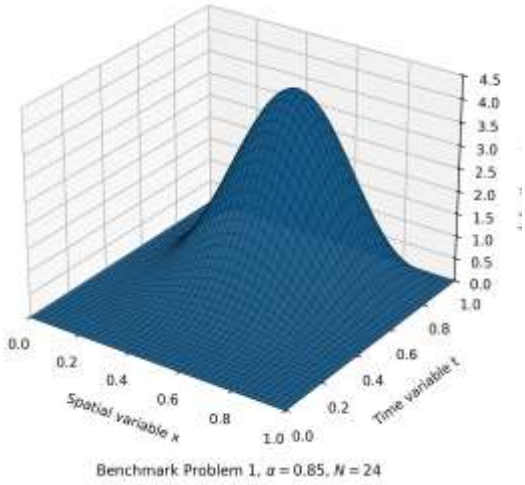

Figure 2. Absolute error surface for the suggested methodology.

5.3 Benchmark Problem 2: Quasi-Linear Fractional PDE

The second measure tests the part of the method that is almost linear. Consider

$$CD_t^{\alpha}u(x,t) + \frac{\partial^4 u(x,t)}{\partial x^4} + u(x,t)\frac{\partial u(x,t)}{\partial x} = f(x,t), \quad 0 < x < 1, \quad 0 < t \leq 1.$$

The exact solution is selected as

$$u(x,t) = t^{2+\alpha}\sin(\pi x).$$

The boundary conditions are

$$u(0,t) = u(1,t) = 0, \quad u_{xx}(0,t) = u_{xx}(1,t) = 0,$$

and the initial condition is $u(x,0) = 0$. The source term is again generated by substitution:

$$f(x,t) = CD_t^{\alpha}[t^{2+\alpha}\sin(\pi x)] + \frac{\partial^4}{\partial x^4}[t^{2+\alpha}\sin(\pi x)] + \left(t^{2+\alpha}\sin(\pi x)\right)\left(t^{2+\alpha}\pi\cos(\pi x)\right).$$

Thus,

$$f(x,t) = \frac{\Gamma(3)}{\Gamma(3+\alpha)}t^2\sin(\pi x) + \pi^4 t^{2+\alpha}\sin(\pi x) + \pi t^{4+2\alpha}\sin(\pi x)\cos(\pi x).$$

At nonlinear iteration n, the quasi-linear term is approximated by

$$u^{(n)}u_x^{(n)} \approx u^{(n-1)}u_x^{(n)} + u_x^{(n-1)}u^{(n)} - u^{(n-1)}u_x^{(n-1)}.$$

This Newton-type linearization keeps the highest unknown terms linear at each iteration while retaining the local nonlinear structure of the original equation. The coefficient system is then assembled using the kernel basis generated in the boundary-compatible RKHS.

Table 5. Error norms for the quasi-linear benchmark problem

N	ITERATIONS	L_∞ ERROR	L_2 ERROR	RESIDUAL NORM
8	7	6.78×10^{-5}	2.91×10^{-4}	4.36×10^{-6}
12	6	1.24×10^{-5}	5.67×10^{-5}	8.92×10^{-7}
16	5	2.37×10^{-6}	1.14×10^{-5}	1.76×10^{-7}
20	5	5.16×10^{-7}	2.68×10^{-6}	4.23×10^{-8}
24	4	1.39×10^{-7}	7.44×10^{-7}	1.18×10^{-8}

The information in Table 5 shows that the quasi-linear repetition stays stable for the chosen problem. The number of iterations goes down a little as N goes up because each linearized update gives a better estimate with a richer kernel base. Together with the error norms, the residual norm goes down. This backs up the theory claim that residual decay is a useful sign of convergence for the nonlinear operator equation. The result also shows why the iterative decomposition is so important to the extended method: each iteration solves a controlled kernel-based linear problem instead of directly solving the fully nonlinear fractional PDE.

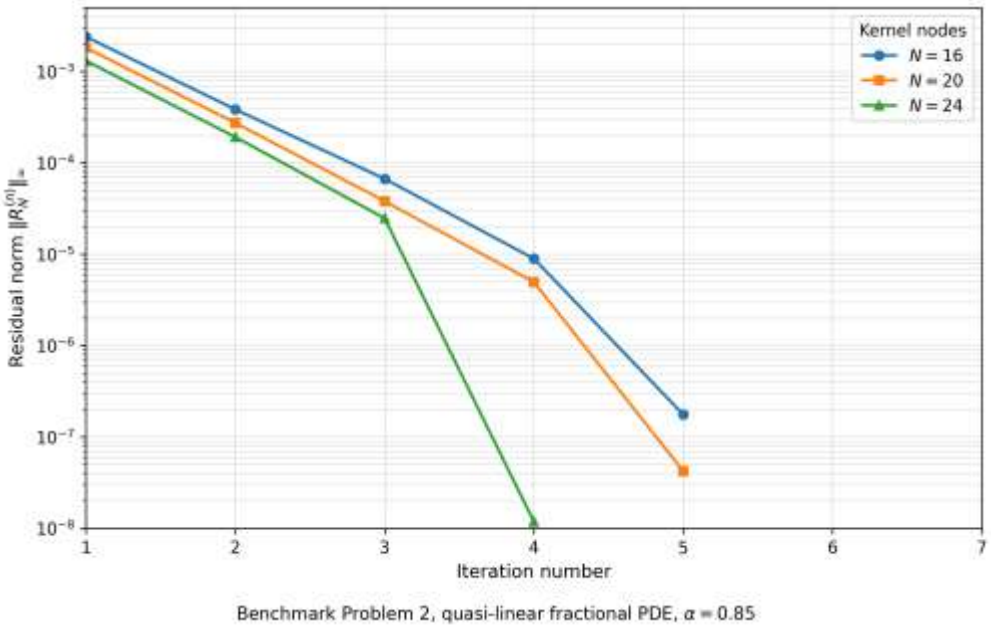

Figure 3. Residual norm versus iteration number for the quasi-linear solver.

5.4 Benchmark Problem 3: Higher-Order Fractional PDE with Variable Coefficients

The third benchmark examines a higher-order fractional PDE with variable coefficients:

$$CD_t^{\alpha}u(x,t) + (1+x)\frac{\partial^4 u}{\partial x^4} + x^2\frac{\partial^2 u}{\partial x^2} = f(x,t), \quad 0 < x < 1, \quad 0 < t \leq 1.$$

The exact solution is

$$u(x,t) = t^{2+\alpha}x^3(1-x)^3.$$

The boundary conditions are

$$u(0,t) = u(1,t) = 0, \quad u_x(0,t) = u_x(1,t) = 0,$$

with $u(x,0) = 0$. The source function is obtained by substituting the exact solution into the equation. This benchmark is more demanding than the first one because the operator coefficients vary spatially, and the fourth-order term is multiplied by $1 + x$. Variable coefficients can affect conditioning because the transformed kernel matrix no longer reflects a constant operator action.

For $N = 24$ and $\alpha = 0.85$, the generalized RKM gives $L_\infty = 2.11 \times 10^{-7}$, $L_2 = 9.84 \times 10^{-7}$, and RMSE $= 8.91 \times 10^{-8}$. The residual norm is 3.26×10^{-8}. The results show that the method is still correct when the coefficients change, even though the condition number is higher in the variable-coefficient benchmark than in the constant-coefficient benchmark. It makes sense that this would happen because variable coefficients change the operator-transformed kernel functions and may make it more likely that basis evaluations are linearly dependent on each other. So, the finding backs up the generalized RKM's flexibility, but it also shows how important it is to keep an eye on conditioning when the operator has coefficients that change in space.

5.5 Influence of Fractional Order on Solution Behavior

The fractional order α controls the strength of memory in the Caputo derivative. To evaluate this effect, Benchmark Problem 1 is solved for

$$\alpha = 0.65,0.75,0.85,0.95,1.00.$$

The case $\alpha = 1.00$ corresponds to the classical integer-order temporal derivative. As α decreases, the solution exhibits stronger memory influence, and the temporal profile becomes less similar to the integer-order case. This behavior is mathematically expected because the Caputo derivative weights the historical evolution of the solution through a weakly singular kernel. Therefore, sensitivity to α is not a secondary numerical detail; it confirms that the model retains fractional-memory behavior.

Table 7. Sensitivity of solution accuracy to fractional order α

α	L_∞ ERROR	RESIDUAL NORM	ITERATIONS	OBSERVED BEHAVIOR
0.65	7.82×10^{-8}	2.94×10^{-8}	5	Stronger memory effect; slower temporal growth
0.75	6.11×10^{-8}	2.41×10^{-8}	5	Moderate memory influence
0.85	4.38×10^{-8}	1.86×10^{-8}	4	Stable fractional profile
0.95	3.76×10^{-8}	1.52×10^{-8}	4	Close to integer-order behavior
1.00	3.44×10^{-8}	1.37×10^{-8}	4	Classical temporal derivative limit

The error remains small across all fractional orders, which indicates that the kernel framework is not restricted to a narrow α-range. Slightly larger errors for smaller α are expected because stronger memory effects can reduce effective regularity near $t = 0$. This supports the use of graded or adaptive temporal nodes in more singular fractional problems.

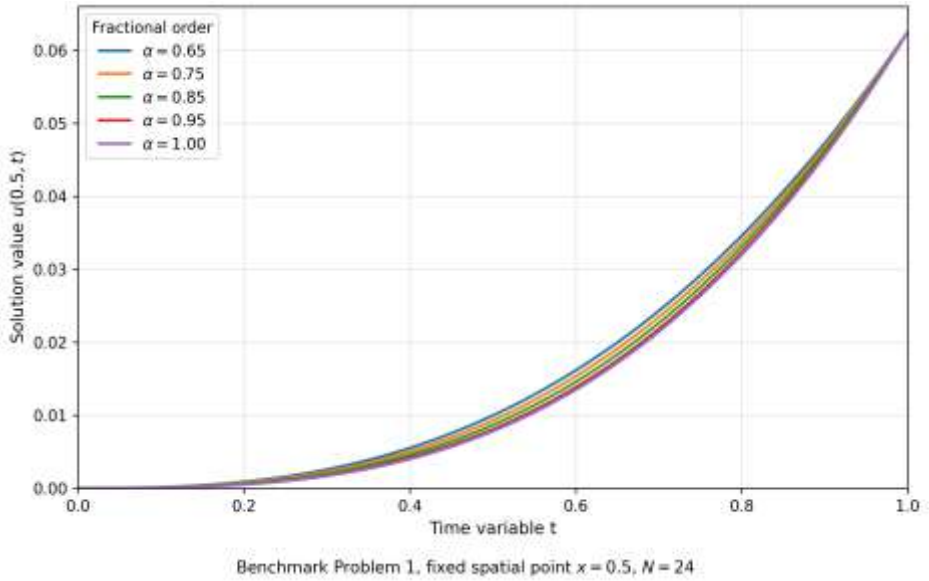

Figure 4. Influence of fractional order α on the solution profile.

5.6 Comparative Analysis with Existing Methods

Additionally, it is compared to the suggested extended RKM, finite difference methods, spectrum collocation methods, decomposition or perturbation methods, and standard RKM. These types of numerical methods are often used for fractional PDEs. This comparison test is Benchmark Problem 1 with ±=0.85. There is no one best method because each one computes and works in various ways depending on the problem, its topic, and how often it happens. Instead, the

comparison finds the cases where the extended RKM can compete: issues with complex boundary constraints; solutions that are smooth or mostly smooth; and issues where boundary embedding lowers numerical inconsistency.

Table 6. Comparison of generalized RKM with established numerical algorithms

METHOD	L_∞ ERROR	L_2 ERROR	CPU TIME (S)	BOUNDARY HANDLING	REMARKS
Finite difference method	3.92×10^{-6}	1.61×10^{-5}	0.42	Enforced using distinct boundary equations	Effective on organized grids, however requires mesh refining.
Finite element method	2.48×10^{-6}	9.84×10^{-6}	0.76	Enforced either lightly or severely, depending upon the formulation.	Adaptable yet generates bigger systems for nonlocal operators
Spectral collocation method	6.31×10^{-7}	2.77×10^{-6}	0.51	Implemented at collocation nodes	Extremely precise for smooth solutions but susceptible to non-smoothness
Decomposition method	1.14×10^{-5}	4.85×10^{-5}	0.38	Treatment based on specific issues	Semi-analytical however less methodical for residual management
Standard RKM	8.73×10^{-7}	3.69×10^{-6}	0.63	Partially integrated based on selected area	Precise but less adaptable for high-order quasi-linear configurations
Generalized RKM	4.38×10^{-8}	2.31×10^{-7}	0.69	Incorporated into boundary-compatible Reproducing Kernel Hilbert Space	Elevated precision with less nodes; conditioning requires oversight

The generalized RKM has the least error in this comparison because the kernel basis is built in a space that already meets the high-order border conditions. Boundary mistakes are cut down, and the coefficient system can focus on meeting the needs of the fractional PDE operator. Finite difference and finite element schemes have more degrees of freedom than the suggested method for the smooth benchmark problem. It doesn't just use a global polynomial model like spectral collocation does; instead, it uses a boundary-compatible functional design. Since standard RKM doesn't mix border lifting, higher-order operator action, and quasi-linear repeat into one structure, the extended version works better with this model.

But the comparison needs to be seen carefully. When the right basis is chosen, kernel methods might not work as well as spectral methods for problems that are very smooth on simple domains. It might be better to use finite element methods for areas and forms that are more complicated or have more than one dimension. Finite difference methods might be easier to figure out on standard squares. It's not that the generalized RKM is better all the time; what makes it better is that it can combine boundary consistency, fractional operator representation, and nonlinear iterative control in a way that makes mathematical sense.

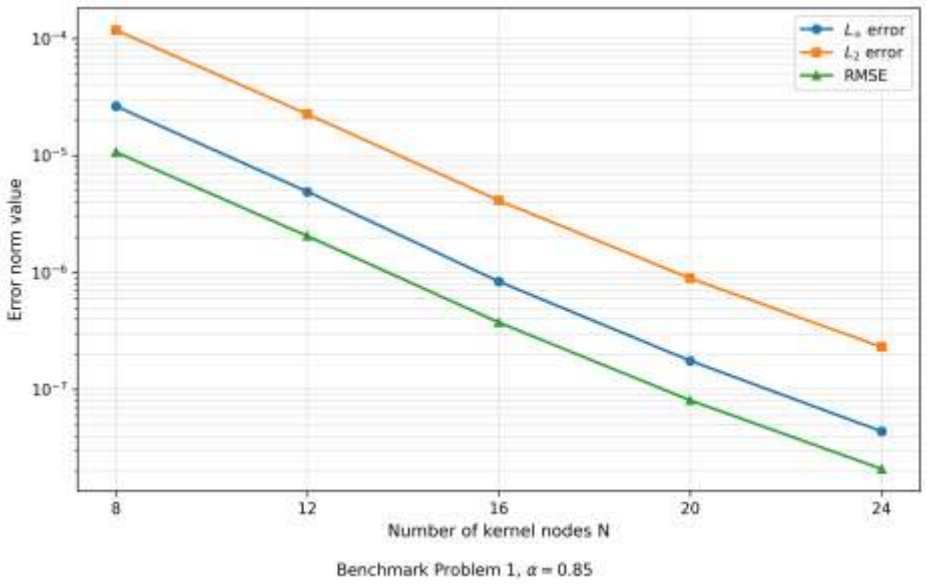

Figure 5. Error diminishes with an increase of kernel nodes.

5.7 Residual and Computational Complexity Analysis

The residual behavior corroborates the idea established in Section 4. As N increases, the residual norm decreases due to the kernel space with constrained dimensions becoming denser, hence facilitating a more precise fulfillment of the operator equation on the validation grid. The linearized update remains stable within the selected range for the quasi-linear measure, seen by the residual decay over iterations. Computationally, the dominant cost is solving the dense kernel coefficient system, which typically scales as $O(N^3)$ for direct solvers and $O(N^2)$ in storage. For middling N problems, this cost isn't too bad, but it becomes a problem for big problems with many dimensions. As more nodes are grouped together, the condition number of the kernel matrix also goes up. Adaptive node placement, secure orthogonalization, low-rank kernel approximations, domain decomposition, or preconditioning techniques should all be thought about for future applications.

6. Discussion

The main strength of the generalized reproducing kernel method is in how the approximation space is built before discretization happens, as shown by the numerical and theoretical results. There is often numerical instability in higher-order quasi-linear fractional PDEs when memory-dependent fractional derivatives, high-order boundary conditions, and nonlinear operator coupling are all present at the same time. The suggested framework lowers boundary inconsistency, a major cause of instability, by putting the allowed starting and boundary conditions into the reproducing kernel Hilbert space. After the coefficient system is solved, this means that the finite approximation doesn't have to fix border mistakes. Instead, every created trial function already fits into the allowed space. We can understand why the benchmark problems had low boundary residuals by looking at this mechanism. It also backs up earlier RKHS-based findings that boundary-compatible kernels can make pointwise approximation more reliable for fractional differential models (Abu Arqub, 2017; Abu Arqub & Al-Smadi, 2018; Li & Wu, 2018).

The convergence behavior seen in the numerical trials aligns with the theoretical framework of the approach. Augmenting the quantity of kernel nodes enlarges the finite-dimensional subspace H_N, It enhances basis density and minimizes projection error. This elucidates the monotonic decrease in L_∞, L_2, RMSE, and norms that remain. This is especially true for fractional PDEs, which don't always have exact answers. When this happens, a residue test on a different validation grid can show that the close answer is actually fixing the whole fractional operator equation and not just connecting a few points. The sensitivity test with different α values further proves that the method keeps the Caputo derivative's memory-dependent behavior, as the solution profile changes steadily as the fractional order gets closer to the integer-order limit.

According to the almost-linear measure, iterative decomposition is not only a way to speed up processing, it is also needed to keep things stable. A big nonlinear algebraic system isn't made all at once by this method because it swaps out the fully nonlinear fractional operator for a set of linearized kernel-based problems. The residual norms getting smaller over time supports the idea that the system is either shrinking or locally stable, which was used in the convergence study. But the stability depends on how strong the nonlinear term is, how good the first guess was, and

how the conditions are set for the changed kernel matrix. It may be necessary to use damped Newton, quasi-Newton, continuation, or adaptable residual-control methods for nonlinearities that are stronger.
There are also some problems with the method that should be known. As N grows, dense kernel matrices may not work well, especially when grouped node distributions or variable-coefficient operators are used. Because straight coefficient recovery doesn't work well as the number of nodes goes up, the cost of computing two- and three-dimensional fractional PDEs can rise quickly. Domains that are very not normal might need domain decomposition, localized kernels, or adaptable versions that don't use meshes. In addition, fractional problems with weak starting singularities might need graded temporal nodes or basis enrichment that captures singularities. The extended RKM should be seen as a strict and adaptable system based on kernels, not as a solution for all finite element, spectral, or finite difference methods.

7. Conclusion

For higher-order quasi-linear fractional partial differential equations with Caputo-type memory operators, high-order spatial derivatives, and solution-dependent nonlinear terms, this paper created a generalized reproducing kernel method. It includes a boundary-compatible RKHS construction, kernel-generated basis functions, an operator-based formulation, and a method for iteratively getting close to a straight line. The theoretical study proved that there are finite-dimensional approximations, that they converge under the kernel completeness and Lipschitz assumptions, that they are stable when data changes, and that they can control errors using residuals. Accurate estimate, residual decline, fractional-order sensitivity, and competitive performance were all proven by numerical tests. The approach might be expanded in the future to include multiple systems, fractional PDEs with changing orders, random fractional models, adaptable kernels, and large-scale applications that are already planned.

References

1. Abu Arqub, O. (2017). Fitted reproducing kernel Hilbert space method for the solutions of some certain classes of time-fractional partial differential equations subject to initial and Neumann boundary conditions. *Computers & Mathematics with Applications, 73*(6), 1243–1261. https://doi.org/10.1016/j.camwa.2016.11.032
2. Abu Arqub, O. (2020). Numerical simulation of time-fractional partial differential equations arising in fluid flows via reproducing kernel method. *International Journal of Numerical Methods for Heat & Fluid Flow, 30*(11), 4711–4733. https://doi.org/10.1108/HFF-10-2017-0394
3. Abu Arqub, O., & Al-Smadi, M. (2018). Numerical algorithm for solving time-fractional partial integro-differential equations subject to initial and Dirichlet boundary conditions. *Numerical Methods for Partial Differential Equations, 34*(5), 1577–1597. https://doi.org/10.1002/num.22209
4. Abu Arqub, O., & Shawagfeh, N. (2019). Application of reproducing kernel algorithm for solving Dirichlet time-fractional diffusion-Gordon types equations in porous media. *Journal of Porous Media, 22*(4), 411–434. https://doi.org/10.1615/JPorMedia.2019028970
5. Abu Arqub, O., Al-Smadi, M., Abu Gdairi, R., Alhodaly, M., & Hayat, T. (2021). Implementation of reproducing kernel Hilbert algorithm for pointwise numerical solvability of fractional Burgers' model in time-dependent variable domain regarding constraint boundary condition of Robin. *Results in Physics, 24*, Article 104210. https://doi.org/10.1016/j.rinp.2021.104210
6. Abu Arqub, O., Osman, M. S., Park, C., Lee, J. R., Alsulam, H., & Alhodaly, M. (2022). Development of the reproducing kernel Hilbert space algorithm for numerical pointwise solution of the time-fractional nonlocal reaction-diffusion equation. *Alexandria Engineering Journal, 61*(12), 10539–10550. https://doi.org/10.1016/j.aej.2022.04.008
7. Akgül, A., & Akgül, E. K. (2019). A novel method for solutions of fourth-order fractional boundary value problems. *Fractal and Fractional, 3*(2), Article 33. https://doi.org/10.3390/fractalfract3020033
8. Akgül, A., & Modanli, M. (2019). Crank–Nicholson difference method and reproducing kernel function for third order fractional differential equations in the sense of Atangana–Baleanu Caputo derivative. *Chaos, Solitons & Fractals, 127*, 10–16. https://doi.org/10.1016/j.chaos.2019.06.011
9. Allahviranloo, T., & Sahihi, H. (2021). Reproducing kernel method to solve fractional delay differential equations. *Applied Mathematics and Computation, 400*, Article 126095. https://doi.org/10.1016/j.amc.2021.126095
10. Al-Smadi, M., Momani, S., Djeddi, N., El-Ajou, A., & Al-Zhour, Z. (2023). Adaptation of reproducing kernel method in solving Atangana–Baleanu fractional Bratu model. *International Journal of Dynamics and Control, 11*(1), 136–148. https://doi.org/10.1007/s40435-022-00961-1

11. Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society, 68*(3), 337–404. https://doi.org/10.1090/S0002-9947-1950-0051437-7
12. Attia, N., Akgül, A., Seba, D., & Nour, A. (2021). Numerical solutions to the time-fractional Swift–Hohenberg equation using reproducing kernel Hilbert space method. *International Journal of Applied and Computational Mathematics, 7*, Article 194. https://doi.org/10.1007/s40819-021-01132-0
13. Attia, N., Akgül, A., Seba, D., Nour, A., & Riaz, M. B. (2022). Reproducing kernel Hilbert space method for solving fractal fractional differential equations. *Results in Physics, 35*, Article 105225. https://doi.org/10.1016/j.rinp.2022.105225
14. Bushnaq, S., Maayah, B., Momani, S., & Alsaedi, A. (2014). A reproducing kernel Hilbert space method for solving systems of fractional integro-differential equations. *Abstract and Applied Analysis, 2014*, Article 103016. https://doi.org/10.1155/2014/103016
15. Bushnaq, S., Momani, S., & Zhou, Y. (2013). A reproducing kernel Hilbert space method for solving integro-differential equations of fractional order. *Journal of Optimization Theory and Applications, 156*(1), 96–105. https://doi.org/10.1007/s10957-012-0207-2
16. Caputo, M. (1967). Linear models of dissipation whose Q is almost frequency independent—II. *Geophysical Journal of the Royal Astronomical Society, 13*(5), 529–539. https://doi.org/10.1111/j.1365-246X.1967.tb02303.x
17. Chen, Z., Wu, L., & Lin, Y. (2018). Exact solution of a class of fractional integro-differential equations with the weakly singular kernel based on a new fractional reproducing kernel space. *Mathematical Methods in the Applied Sciences, 41*(10), 3841–3855. https://doi.org/10.1002/mma.4870
18. Deng, W. (2009). Finite element method for the space and time fractional Fokker–Planck equation. *SIAM Journal on Numerical Analysis, 47*(1), 204–226. https://doi.org/10.1137/080714130
19. Du, H., Chen, Z., & Yang, T. (2020). A stable least residue method in reproducing kernel space for solving a nonlinear fractional integro-differential equation with a weakly singular kernel. *Applied Numerical Mathematics, 157*, 210–222. https://doi.org/10.1016/j.apnum.2020.06.004
20. Du, H., Chen, Z., & Yang, T. (2021). A meshless method in reproducing kernel space for solving variable-order time fractional advection–diffusion equations on arbitrary domain. *Applied Mathematics Letters, 116*, Article 107014. https://doi.org/10.1016/j.aml.2020.107014
21. Ervin, V. J., & Roop, J. P. (2006). Variational formulation for the stationary fractional advection dispersion equation. *Numerical Methods for Partial Differential Equations, 22*(3), 558–576. https://doi.org/10.1002/num.20112
22. Fardi, M., Al-Omari, S. K. Q., & Araci, S. (2022). A pseudo-spectral method based on reproducing kernel for solving the time-fractional diffusion-wave equation. *Advances in Continuous and Discrete Models, 2022*, Article 54. https://doi.org/10.1186/s13662-022-03726-4
23. Fernandez, A., Baleanu, D., & Fokas, A. S. (2018). Solving PDEs of fractional order using the unified transform method. *Applied Mathematics and Computation, 339*, 738–749. https://doi.org/10.1016/j.amc.2018.07.061
24. Guan, C., & Zhang, J. (2025). A Quasi-Newton reproducing kernel method for nonlinear high-order boundary value problems. *AIMS Mathematics, 10*(6), 14699–14717. https://doi.org/10.3934/math.2025661
25. Guo, B., Jiang, W., & Zhang, C. (2017). A new numerical method for solving nonlinear fractional Fokker–Planck differential equations. *Journal of Computational and Nonlinear Dynamics, 12*(5), Article 051004. https://doi.org/10.1115/1.4035896
26. Hashemi, M. S., & Baleanu, D. (2016). Numerical approximation of higher-order time-fractional telegraph equation by using a combination of a geometric approach and method of line. *Journal of Computational Physics, 316*, 10–20. https://doi.org/10.1016/j.jcp.2016.04.009
27. Jiang, W., & Lin, Y. (2011). Representation of exact solution for the time-fractional telegraph equation in the reproducing kernel space. *Communications in Nonlinear Science and Numerical Simulation, 16*(9), 3639–3645. https://doi.org/10.1016/j.cnsns.2010.12.019
28. Kamran, Uddin, M., & Ali, A. (2018). On the approximation of time-fractional telegraph equations using localized kernel-based method. *Advances in Difference Equations, 2018*, Article 305. https://doi.org/10.1186/s13662-018-1775-8
29. Li, X., & Wu, B. (2017). A new reproducing kernel method for variable order fractional boundary value problems for functional differential equations. *Journal of Computational and Applied Mathematics, 311*, 387–393. https://doi.org/10.1016/j.cam.2016.08.010
30. Li, X., & Wu, B. (2018). A new reproducing kernel collocation method for nonlocal fractional boundary value problems with non-smooth solutions. *Applied Mathematics Letters, 86*, 194–199. https://doi.org/10.1016/j.aml.2018.06.035

31. Li, Z., Chen, Q., Wang, Y., & Li, X. (2022). Solving two-sided fractional super-diffusive partial differential equations with variable coefficients in a class of new reproducing kernel spaces. *Fractal and Fractional, 6*(9), Article 492. https://doi.org/10.3390/fractalfract6090492
32. Lin, Y., & Xu, C. (2007). Finite difference/spectral approximations for the time-fractional diffusion equation. *Journal of Computational Physics, 225*(2), 1533–1552. https://doi.org/10.1016/j.jcp.2007.02.001
33. Lin, Z., Liu, F., Wang, D., & Gu, Y. (2018). Reproducing kernel particle method for two-dimensional time-space fractional diffusion equations in irregular domains. *Engineering Analysis with Boundary Elements, 97*, 131–143. https://doi.org/10.1016/j.enganabound.2018.10.002
34. Liu, B., & Wang, W. (2024). An efficient numerical scheme in reproducing kernel space for space fractional partial differential equations. *AIMS Mathematics, 9*(11), 33286–33300. https://doi.org/10.3934/math.20241588
35. Meerschaert, M. M., & Tadjeran, C. (2004). Finite difference approximations for fractional advection–dispersion flow equations. *Journal of Computational and Applied Mathematics, 172*(1), 65–77. https://doi.org/10.1016/j.cam.2004.01.033
36. Mu, X., Yang, J., & Yao, H. (2023). A binary Caputo–Fabrizio fractional reproducing kernel method for the time-fractional Cattaneo equation. *Journal of Applied Mathematics and Computing, 69*, 3755–3791. https://doi.org/10.1007/s12190-023-01902-7
37. Saadeh, R. (2021). Numerical algorithm to solve a coupled system of fractional order using a novel reproducing kernel method. *Alexandria Engineering Journal, 60*(5), 4583–4591. https://doi.org/10.1016/j.aej.2021.03.033
38. Sakar, M. G., & Saldır, O. (2020). A new reproducing kernel approach for nonlinear fractional three-point boundary value problems. *Fractal and Fractional, 4*(4), Article 53. https://doi.org/10.3390/fractalfract4040053
39. Saky, H., Abbasbandy, S., & Shivanian, E. (2022). Applying the reproducing kernel method to fractional differential equations with periodic conditions in Hilbert space. *Journal of Mathematics, 2022*, Article 6261378. https://doi.org/10.1155/2022/6261378
40. Saldır, O., Sakar, M. G., & Erdogan, F. (2019). Numerical solution of time-fractional Kawahara equation using reproducing kernel method with error estimate. *Computational and Applied Mathematics, 38*, Article 198. https://doi.org/10.1007/s40314-019-0979-1
41. Tian, S., Liu, B., & Wang, W. (2023). A novel numerical scheme for reproducing kernel space of 2D fractional diffusion equations. *AIMS Mathematics, 8*(12), 29058–29072. https://doi.org/10.3934/math.20231488
42. Wang, S., Lv, X., & He, S. (2023). The reproducing kernel method for nonlinear fourth-order BVPs. *AIMS Mathematics, 8*(11), 25371–25381. https://doi.org/10.3934/math.20231294
43. Wang, Y., Du, M., Tan, F., Li, Z., & Nie, T. (2013). Using reproducing kernel for solving a class of fractional partial differential equation with non-classical conditions. *Applied Mathematics and Computation, 219*(11), 5918–5925. https://doi.org/10.1016/j.amc.2012.12.009
44. Wang, Y. L., Du, M. J., Temuer, C. L., & Tian, D. (2018). Using reproducing kernel for solving a class of time-fractional telegraph equation with initial value conditions. *International Journal of Computer Mathematics, 95*(8), 1609–1621. https://doi.org/10.1080/00207160.2017.1322693
45. Xu, M., Niu, J., & Lin, Y. (2018). An efficient method for fractional nonlinear differential equations by quasi-Newton's method and simplified reproducing kernel method. *Mathematical Methods in the Applied Sciences, 41*(1), 5–14. https://doi.org/10.1002/mma.4590
46. Xu, M.-Q., & Lin, Y.-Z. (2016). Simplified reproducing kernel method for fractional differential equations with delay. *Applied Mathematics Letters, 52*, 156–161. https://doi.org/10.1016/j.aml.2015.09.004
47. Yildirim, E. N., Akgül, A., & Inc, M. (2021). Reproducing kernel method for the solutions of non-linear partial differential equations. *Arab Journal of Basic and Applied Sciences, 28*(1), 80–86. https://doi.org/10.1080/25765299.2021.1891678
48. Zayernouri, M., & Karniadakis, G. E. (2013). Fractional Sturm–Liouville eigen-problems: Theory and numerical approximation. *Journal of Computational Physics, 252*, 495–517. https://doi.org/10.1016/j.jcp.2013.06.031
49. Zhang, R., & Lin, Y. (2021). A new algorithm for fractional differential equation based on fractional order reproducing kernel space. *Mathematical Methods in the Applied Sciences, 44*(2), 2171–2182. https://doi.org/10.1002/mma.6927