Miskolc Mathematical Notes
Vol. 26 (2025), No. 2, pp. 767–783

HU e-ISSN 1787-2413
DOI: 10.18514/MMN.2025.4965

# LS-SVR FOR THE FRACTIONAL FOURTH-ORDER PARTIAL DIFFERENTIAL EQUATIONS

MARZIYEH FELAHAT

*Received 16 February, 2024*

*Abstract.* Fractional fourth-order partial differential equations find various applications such as image denoising, electrostatics, and geometric modeling. In this paper, we propose a new supervised machine learning algorithm for such problems. To do so, we develop a simulation method based on least squares support vector regression (LS-SVR). A polynomial kernel is used to approximate the solution in the training process by considering an inverse viewpoint to the residual function. For minimizing the loss function, we use the Petrov-Galerkin method for the constraints in the proposed LS-SVR. We study the stability and convergence of the method by providing some numerical examples and illustrating the error behavior as the degree of the solution increases.

2010 *Mathematics Subject Classification:* 35R11; 65N30; 65N22; 65M60

*Keywords:* fourth-order PDE, fractional PDEs, least squares support vector regression, orthogonal kernel

## 1. INTRODUCTION

Fractional partial differential equations (FPDEs) are used to model a variety of physical phenomena in fluid dynamics, material science, and chemical engineering such as anomalous diffusion, wave propagation, and viscoelasticity. Especially, fourth-order FPDEs play an important role in the study of phase transitions and chaotic phenomena in complex systems. Furthermore, they can be used to represent the dynamics of some nonlinear systems, such as the long time behavior for population models. By modeling physical phenomena with FPDEs, it is possible to obtain a more accurate description of the system, which can improve the prediction and decision-making. On the other hand, FPDEs are typically more difficult to solve than traditional partial differential equations, so they require more specific numerical methods to handle the singularity in the kernel of the integral in the fractional derivative [2, 31, 34, 36].

Classic numerical methods have been implemented for solving fourth-order FPDEs, including the finite difference [10, 41], finite element [8, 18, 19], finite volume [3, 9, 30], and the Spectral method [11, 15, 17]. The choice of method depends on the specific problem and desired accuracy. The Finite difference is widely used and known for its simplicity, as it involves discretizing the equation and solving the resulting algebraic system. Finite element is more accurate and efficient for complex geometries, while finite volume suits non-uniform grids. The Spectral methods are highly accurate but computationally expensive.

On the other hand, machine learning (ML), neural networks, and deep learning algorithms have become increasingly popular for a wide range of problems in recent years, for instance in pattern recognition, recommender systems, health care, theorem proving, cybersecurity, financial services, and more. ML algorithms use some training data to recognize the objects in groups, detect similarities and give the target in classification, clustering, and regression problems [21].

Recently, ML techniques have been developed to approximate the solution of ordinary, partial, and integral equations (IEs). For instance, Mall et al. [26] presented Chebyshev neural networks for solving Lane-Emden type equations. Lu et al. [25] used deep learning to solve some PDEs numerically. Jafarian et al. [25] used neural networks to solve IEs and Mall et al. [15] used radial basis function neural networks to solve elliptic PDEs. These works show the efficiency of ML for solving complex mathematical problems, especially differential and integral equations.

Support Vector Machines (SVM) is a well-known learning algorithm that was first introduced by Cortes and Vapnik in 1995 for classification problems [6]. It was initially designed as a binary classification algorithm. The SVM model has since been further refined and is now used for a variety of classification and regression tasks. The original goal of the SVM algorithm was to find a maximum margin hyperplane that could separate two sets of data points in a high-dimensional space using kernel functions [39].

Support Vector Regression (SVR) was first proposed by Cortes and Vapnik in 1995 as a revision of SVM for regression problems. The main idea behind SVR is to find a function or hyperplane that best fits the data while keeping the error within a certain margin, called the epsilon-insensitive tube, a range around the true target values where errors are considered acceptable and do not contribute to the loss function [38]. Least Squares Support Vector Machines (LS-SVM) is an extension of the SVM algorithm introduced by Suykens et al. that transforms the problem into a convex optimization task with equality constraints, so it is faster for large datasets and complex regression problems. A revision of this algorithm for regression tasks, LS-SVR has been also extended for solving ODEs [25, 27], PDEs [1, 28], IEs [12, 32, 35], and differential equations on unbounded domains [5, 24].

The superior performance of these methods has motivated us to develop an ML method for solving fourth-order FPDEs. To present the problem and the method, we need some notations.

In this article, the fourth-order FPDE with Caputo time fractional derivative is considered as

$$\partial_t^\alpha u(x,t) + \partial_x^4 u(x,t) + \eta u(x,t) = f(x,t), \qquad x \in [0,1], \ \ t \in [0,T] \tag{1.1}$$

subject to the following initial condition

$$u(x,0) = g(x), \qquad x \in [0,1] \tag{1.2}$$

and boundary conditions

$$u(0,t) = 0, u(1,t) = 0, u_{xx}(0,t) = 0, u_{xx}(1,t) = 0, \qquad t \in [0,T] \tag{1.3}$$

where parameter $0 < \alpha \leq 1$ describes the order of the fractional time derivatives in the Caputo sense given in Eq (1.4), $u(x,t)$ is the unknown function which represents the concentration of the drug at the location $x$ in the tissue at time $t$, in biological and drug release applications, $T$ is the final time at a given period, $\eta$ is a nonnegative constant, and $f(x,t)$ is the given source function [16, 31].

The fractional derivative has been defined in many different ways, Caputo and Riemann-Liouville definitions are frequently used in modeling and applications [4, 29]. Caputo's derivative seems to be more suitable to model some physical phenomena, especially in anomalous diffusion in crowded environments, because it allows boundary and initial conditions to be included in the formulation of the problem. The Caputo time-fractional derivative of order $0 < \alpha \leq 1$ denoted by $\partial_t^\alpha u(x,t)$ in Eq (1.1), is defined by

$$\partial_t^\alpha u(x,t) = \begin{cases} \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\frac{\partial u}{\partial t}(x,s)}{(t-s)^\alpha} ds, & 0 < \alpha < 1, \\ \partial_t u(x,t), & \alpha = 1. \end{cases} \tag{1.4}$$

The novelty of this work is to develop an LS-SVR algorithm with a polynomial kernel based on Petrov-Galerkin for the numerical solution of Eq (1.1)-(1.3).

The rest of the paper is organized as follows. Some preliminaries and notations for polynomial spaces, orthogonal kernels, and SVMs are presented in Section 2. Least squares support vector regression for a general differential operator is then presented in Section 3. Section 4 is devoted to the development of a LS-SVR algorithm for the fractional FPDEs (1.1)-(1.3). Furthermore, numerical examples are provided in Section 5 to demonstrate the accuracy of the proposed method.

## 2. PRELIMINARIES

In this section, we collect some results on the orthogonal polynomials and the least squares support vector regression required later in the formulation of the proposed method.

2.1. *functional spaces and function approximation*

We begin by introducing some notations about functional spaces that will be adopted in the subsequent discussion [13, 21, 24].

Let $\Omega \subset \mathbb{R}^n$, $C(\Omega)$ and $C^k(\Omega)$ denote the set of all continuous functions and functions with continuous derivative up to order $k$ on $\Omega$, respectively. $L^2$ be the set of square integrable functions on $\Omega$ such that

$$\int_\Omega |f|^2 d\mu < \infty,$$

with the standard inner product

$$\langle f, g\rangle = \int_\Omega f g d\mu.$$

**Definition 1.** [21] Let $\Omega \subset \mathbb{R}^n$ be an open set, $m$ be a positive integer and $1 \leq p \leq \infty$. The sobolev space $W^{m,p}(\Omega)$ is defined by

$$W^{m,p}(\Omega) = \{u \in L^p(\Omega) \mid D^\alpha u \in L^p(\Omega), \ \ \forall |\alpha| \leq m\},$$

endowed with the norm $\|\cdot\|_{m,p,\Omega}$ as

$$\|u\|_{m,p,\Omega} = \Big( \sum_{|\alpha|\leq m} \|D^\alpha u\|^p_{L^p(\Omega)} \Big)^{\frac{1}{p}}$$

if $1 \leq p < \infty$ and

$$\|u\|_{m,p,\Omega} = \max_{|\alpha|\leq m} \|D^\alpha u\|_{L^\infty(\Omega)}.$$

Let $w(x)$ be a positive weight function on $\Omega = [0,1]$. We use the weighted Sobolev spaces $H^k_w(\Omega) = W^{k,2}(\Omega)$ whose norm is denoted by $\|\cdot\|_{k,w}$. To accommodate homogeneous boundary conditions, we introduce the following definition:

$$H^2_{0,w}(\Omega) = \{v \in H^2_w(\Omega) : v(0) = v(1) = v'(0) = v'(1) = 0\},$$

where $p = \infty$.

Function approximation can be achieved through various methods such as least squares approximation, interpolation, and fitting. Low degree polynomials are the most widely used functions in approximation theory, due to the easy computations as well as smoothness properties. Moreover, they provide a guaranteed convergence for approximating continuous functions on compact sets as stated by the Weierstrass theorem.

Orthogonal bases allow us to transform the complicated functional equations into algebraic equations that are computationally efficient. As the most common orthogonal functions, the Jacobi polynomials $J_n^{\alpha,\beta}(x)$ are introduced as the eigenfunctions $u(x)$ of Sturm-Liouville differential equation [20, 37],

$$-(1-x)^{-\alpha}(1+x)^{-\beta}\frac{d}{dx}\left((1-x)^{\alpha+1}(1+x)^{\beta+1}\frac{d}{dx}u(x)\right) \tag{2.1}$$

$$= (x^2-1)\frac{d^2}{dx^2}u(x) + (\alpha-\beta+(\alpha+\beta+2)x)\frac{d}{dx}u(x),$$

with $\alpha,\beta > -1$. When $\alpha = \beta = 0$, the Jacobi polynomials reduce to a widely used, the Legendre polynomials $P_n(x)$, $n = 0,1,\cdots$. These polynomials are orthogonal as expressed by the inner product

$$\langle P_n(x), P_m(x)\rangle = \frac{2}{2n+1}\delta_{mn},$$

where $\delta_{mn}$ denotes the Kronecker delta and $\langle\cdot,\cdot\rangle$ being the standard inner product of two functions on the interval $[-1,1]$. The completeness and orthogonality of these polynomials are further discussed within the framework of Sturm-Liouville theory. Eq (2.1) is simplified for the Legendre polynomials as

$$\frac{d}{dx}((1-x^2)\frac{d}{dx}P_n(x)) + \lambda_n P_n(x) = 0,$$

with the eigenvalues $\lambda_n = n(n+1)$. An explicit formula is given by [37]

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx}(x^2-1)^n, \qquad n \geq 0.$$

However, it requires higher order derivatives. A simple recursive formula is given by

$$P_0(x) = 1, \quad P_1(x) = x,$$
$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \qquad n \geq 1.$$

These polynomials enjoy some computational advantages like the following symmetry and boundedness relations

$$P_n(-x) = (-1)^n P_n(x), \quad n \geq 0, \tag{2.2}$$
$$|P_n(x)| \leq 1, \qquad \forall x \in [-1,1], \tag{2.3}$$

where relation (2.2) reduces the computations and (2.3) keeps the error propagation small.

### 2.2. *Support vector regression*

Let us consider a given training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^d$ as the training input data and $y_i \in \mathbb{R}$ as the corresponding output. In ε-SVR, the goal is to find a function $u(x)$ that has at most ε deviation from the obtained targets $y_i$ for all the training data, and at the same time is as flat as possible [38, 40]. The unknown solution $u(x)$ is expanded in terms of a weighted summation over some basis functions

$$u(x) = \mathbf{w}^T\varphi(x) + b.$$

To find $\mathbf{w}$ and $b$, the following primal optimization problem is solved [7]

$$\min_{\mathbf{w},b,\varepsilon} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{\gamma}{2}\varepsilon^T\varepsilon \tag{2.4}$$
$$\text{s.t.} \quad y_i = \mathbf{w}^T\varphi(x_i) + b + \varepsilon_i, \quad i = 1,\cdots,n.$$

where $\mathbf{w} = [w_1, \cdots, w_d]^T$ is a weight vector in $\mathbb{R}^d$ and $b \in \mathbb{R}$ is the bias term, $\gamma$ is the regularization parameter that can be related to the Tikhonov regularization, $\varepsilon = [\varepsilon_1, \cdots, \varepsilon_n]^T$ is error vector where $\varepsilon_i$ is the error of the $i$th training point. The vector valued function $\varphi(x) \in \mathbb{R}^d$ can be a linear or nonlinear function that maps an input space $\mathbb{R}^d$ into a higher dimensional feature space. The following theorem gives the approximate LS-SVR solution by solving a linear system of equations with explicitly given matrices. For the support vector machines, we consider dual space and use the Lagrangian primal method. To find the optimum of the constraint function in the dual space, we have to use the Karush-Kuhn-Tucker (KKT) conditions (Karush 1939; Kuhn and Tucker 1951; Fletcher 1987).

**Theorem 1.** *The solution of quadratic programming* (2.4) *is uniquely given by the following linear system,*

$$\left[\begin{array}{c|c} 0 & \mathbf{1}_n^T \\ \hline \mathbf{1}_n & \Omega + \frac{I}{\gamma} \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathbf{y} \end{array}\right] \tag{2.5}$$

*where $I$ is the identity matrix, $\Omega$ a positive definite matrix, $\Omega_{ij} = \varphi(x_i)^T \varphi(x_j)$ for $i, j = 1, \cdots, n$, $\mathbf{y} = [y_1, \cdots, y_n]^T$, $\mathbf{1}_n = [1, \cdots, 1]^T$ and the dual variables $\alpha = [\alpha_1, \cdots, \alpha_n]^T$. Then, for the kernel $K(x, x_i) = \varphi(x)^T \varphi(x_i)$, we get the solution as*

$$u(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i) + b. \tag{2.6}$$

*Proof.* The proof is straightforward and can be found in [39]. □

In the primal form of LS-SVR, we apply kernel ridge regression to find an unknown function $u(x)$ so we can interpret this as a single hidden layer feed-forward network extended by the $L^2$-regularization technique. Such a network can be seen in Figure 1.

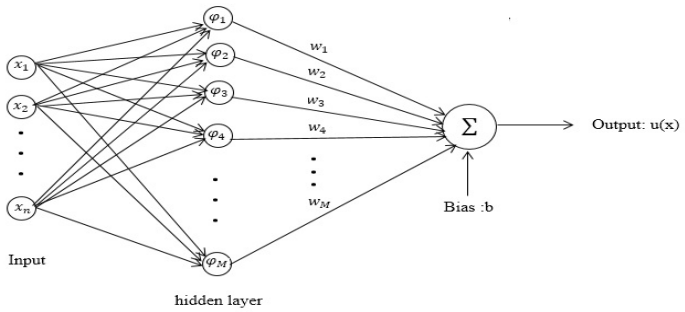


FIGURE 1. The function approximation model used in LS-SVR as a network, $u(x) = \mathbf{w}^T \varphi(x) + b$.

## 3. LS-SVR FORMULATION FOR A GENERAL DIFFERENTIAL OPERATOR

LS-SVR is a powerful and efficient method for solving functional equations. In this method, a set of input data points are mapped into a higher dimensional space, then solves a linear regression problem to make predictions for the new data points which use the least squares estimate to minimize the cost function.

We first present LS-SVR formulation for a general functional equation. Consider $\mathcal{L}u = f$ on a spatial domain $\Omega$ with some boundary or initial conditions $Bu(x_i) = u_i$ where $\mathcal{L}$ may be a differential, integral or integro differential operator. To solve this problem by a learning algorithm, we expand the unknown solution $u(x)$ in terms of a weighted summation over some basis functions as:

$$u(x) \simeq \tilde{u}(x) = \mathbf{w}^T \varphi(x), \tag{3.1}$$

where $\varphi = [\varphi_1, \cdots, \varphi_M]^T$ are basis functions, $\mathbf{w} = [w_1, \cdots, w_M]^T$ are unknown coefficients vector. To determine the unknown weight coefficients $w_j$ in Eq (3.1), we consider the following form of optimization problem associated with Eq (3.1)

$$\begin{aligned} \min_{\mathbf{w},\varepsilon} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{\gamma}{2}\varepsilon^T\varepsilon \\ \text{s.t.} \quad & \left\langle w^T \mathcal{L}\phi(x), \psi_i \right\rangle - \langle f(x), \psi_i \rangle = \varepsilon_i, \quad i = 1, \cdots, n, \end{aligned} \tag{3.2}$$

where $n$ is the number of training data point $x_i$ in $\Omega$, $\varepsilon_i$ is the redidual at $i$th training point, $\gamma$ is the regularization parameter and $\psi_i$ the test functions. By using the Lagrangian multipliers the problem is solved in dual space leading to a linear or nonlinear system of algebraic equations based on the linearity of operator $\mathcal{L}$.

*Remark* 1. In this formulation, we assumed a steady state problem so we only have conditions on space. These conditions always can be assumed to be homogeneous without loss of generality. The nonhomogeneous problems are first converted to homogeneous ones. The basis functions are assumed to satisfy the homogeneous boundary conditions.

**Theorem 2.** *The solution of problem* (3.2) *is uniquely given by the following linear system,*

$$\left(\Omega + \frac{1}{\gamma} I\right) \alpha = \mathbf{y}, \tag{3.3}$$

*where* $\Omega_{ik} = \sum_{k=0}^{n} \left\langle \mathcal{L}\phi_j, \psi_i \right\rangle \left\langle \mathcal{L}\phi_j, \psi_k \right\rangle$ *for* $k, i = 0, \cdots, n$, $\mathbf{y}_i = \langle f, \psi_i \rangle$ *and* $\alpha = [\alpha_1, \cdots, \alpha_n]^T$ *are unknown Lagrange multipliers and*

$$u(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i).$$

*Proof.* The proof is straightforward and can be found in [33]. □

*Remark* 2. Note that the system (3.3) is simpler than the linear system (2.5) for known data which was in a block matrix. This is due to the inclusion of bias in the basis functions.

## 4. Solving fourth-order FPDEs with LS-SVR

In this section, we present the LS-SVR for fourth-order FPDE (1.1)-(1.3) using an orthogonal polynomial kernel. We first discretize the problem in time by using $L^1$ discretization for the Caputo derivative $\partial_t^\alpha u$ [14, 22, 23].

Let $t_0 = 0 < t_1 < \cdots < t_N = T$ be a partition of $[0,T]$. By the Caputo derivative (1.4), in time for function $u(x,t)$ at $t = t_{n+1}, n = 0,1,\ldots$ is given by:

$$\partial_t^\alpha u\Big|_{t=t_{n+1}} = \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{n} \int_{t_j}^{t_{j+1}} \frac{\frac{\partial u}{\partial t}(x,s)}{(t_{n+1}-s)^\alpha} ds$$

by approximating $\frac{\partial u}{\partial t}(x,s)$ with a finite difference on the end points, taking $\tau = t_{j+1} - t_j$, and computing the integrals exactly, we have:

$$\partial_t^\alpha u\Big|_{t=t_{n+1}} \approx \frac{1}{\tau\Gamma(2-\alpha)} \sum_{j=0}^{n} (u^{j+1} - u^j)\Big[(n+1-j)^{1-\alpha} - (n-j)^{1-\alpha}\Big]. \tag{4.1}$$

Assume $a_j^\alpha = (n+1-j)^{1-\alpha} - (n-j)^{1-\alpha}$. Also, if we take $a_{-1}^\alpha = 0$ and $a_{n+1}^\alpha = 0$, Eq (4.1) can be rewritten as:

$$\partial_t^\alpha u\Big|_{t=t_{n+1}} \approx \frac{1}{\tau^\alpha\Gamma(2-\alpha)} \sum_{j=0}^{n+1} \big(a_{j-1}^\alpha - a_j^\alpha\big) u^j \tag{4.2}$$

By taking $\mu_\tau^\alpha := \frac{1}{\tau^\alpha\Gamma(2-\alpha)}$, $b_j^\alpha := a_{j-1}^\alpha - a_j^\alpha$ and $b_{n+1}^\alpha = 1$ in Eq (4.2), we get:

$$\partial_t^\alpha u\Big|_{t=t_{n+1}} \approx \mu_\tau^\alpha u^{n+1} + \mu_\tau^\alpha \sum_{j=0}^{n} b_j^\alpha u^j. \tag{4.3}$$

This is an discrete approximation of fractional derivative as a weighted sum of function values from $t_0$ to $t_{n+1}$.

Now, for the fourth-order FPDE (1.1)-(1.3), suppose $u^0, u^1, \cdots, u^n$ are known and the aim is to find an approximation for $u(x,t)$ at $t = t_{n+1}$. In order to solve the problem approximately with LS-SVR, we apply Crank-Nicolson scheme for $\theta = \frac{1}{2}$ to Eq (4.3), so we get:

$$\mu u^{n+1} + \sum_{j=0}^{n} b_j u^j + \frac{1}{2}\big(\partial_x^4 u^{n+1} + \eta u^{n+1} - f^{n+1}\big) + \frac{1}{2}\big(\partial_x^4 u^n + \eta u^n - f^n\big) = 0. \tag{4.4}$$

Let us assume that the general approximate solution to Eq (4.4) in $t = t_{n+1}$ is of the following form with the weights trained during the learning process

$$u_M^{n+1}(x) \simeq \sum_{j=0}^{M} w_j^{n+1} \varphi_j(x) = \mathbf{w}^T \varphi(x), \tag{4.5}$$

where $M+1$ is the number of basis functions, $\varphi = [\tilde{\varphi}_0, \cdots, \tilde{\varphi}_M]^T$ is the vector of orthogonal basis function that we considered to be modal shifted Legendre polynomials as:

$$\begin{aligned} &\tilde{\varphi}_i(x) = P_i(x) + a_i P_{i+1}(x) + b_i P_{i+2}(x) + c_i P_{i+3}(x) + d_i P_{i+4}(x), \\ &\varphi_i(x) = \tilde{\varphi}_i(2x-1), \qquad i = 0, 1, \cdots, M, \end{aligned} \tag{4.6}$$

where the weights of the polynomials are enforced such that the homogeneous conditions are satisfied, $a_i$, $b_i$, $c_i$ and $d_i$ are updated for any $M$ by imposing $\varphi_i(0) = 0$, $\varphi_i(1) = 0$, $\varphi_i'(0) = 0$ and $\varphi_i'(1) = 0$, and $\mathbf{w} = [w_0, \cdots, w_M]^T$ are unknown weights to be determined in the learning process as described below. In Eq (4.5), for the sake of simplicity, we omit the superscript in $u_M^{n+1}(x)$ and $w_j^{n+1}$. So by replacing (4.5) in (4.4) we have:

$$\mu u^{n+1} + \frac{1}{2}\partial_x^4 u^{n+1} + \frac{1}{2}\eta u^{n+1} = \frac{1}{2}(f^{n+1} - f^n) - \frac{1}{2}(\partial_x^4 u^n + \eta u^n) - \sum_{j=0}^{n} b_j u^j. \tag{4.7}$$

This is written as

$$\mathcal{L} u^{n+1}(x) = F^n(x) \qquad n = 0, 1, \cdots, N, \tag{4.8}$$

where $\mathcal{L}$ is the operator on the left and $F^n(x)$ is the right side of Eq (4.7).

*Remark* 3. The weights $w_j$ in Eq (4.5) are updated at each time step by (4.14) and (4.19).

The variational formulation for Eq (4.7) is given by

$$a(u, v) = \langle F^n, v \rangle, \qquad \forall v \in H_{0,w}^2(\Omega), \tag{4.9}$$

where

$$a(u, v) := \mu \langle u^{n+1}, v \rangle + \frac{1}{2} \langle \partial_x^2 u, \partial_x^2 v \rangle + \frac{1}{2} \eta \langle u^{n+1}, v \rangle, \quad \forall v \in H_{0,w}^2(\Omega),$$

Since $a(u, v)$ is a continuous coercive bilinear form in $H_{0,w}^2(\Omega) \times H_{0,w}^2(\Omega)$ and $\langle F^n, v \rangle$ is a bounded linear function, the Lax-Milgram Lemma [30], guarantees the existence of solution for Eq (4.9).

For the method to be practical, we choose a finite dimensional subspace of $H_{0,w}^2(\Omega)$ as $W_{0,M}(\Omega)$, including the polynomials with the degree at most $M$ satisfying the

boundary conditions (1.3) and seek the approximate solution in $W_{0,M}(\Omega)$ as Eq (4.5) such that

$$\mu\left\langle u_M^{n+1}, v_M\right\rangle + \frac{1}{2}\left\langle \partial_x^2 u_M^{n+1}, \partial_x^2 v_M\right\rangle + \frac{1}{2}\eta\left\langle u_M^{n+1}, v_M\right\rangle = \left\langle F^n, v_M\right\rangle, \quad \forall v_M \in W_{0,M}(\Omega). \tag{4.10}$$

Note that if in (4.10), we set $v_M = \delta(x - x_i)$ for some learning points on the spatial interval, then the primal form of LS-SVR for Eq (4.8) is equivalent to

$$\begin{aligned} \min_{\mathbf{w},\varepsilon} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{\gamma}{2}\varepsilon^T\varepsilon \\ \text{s.t.} \quad & \mathcal{L}u^{n+1}(x_i) - F^n(x_i) = \varepsilon_i, \quad i = 1, \cdots, n \end{aligned} \tag{4.11}$$

with $\mathcal{L}u = \mu u + \frac{1}{2}\partial_x^4 u + \frac{1}{2}\eta u$. This is Collocation LS-SVR for the fourth-order FPDE (1.1)-(1.3). This is well established, however, to take advantage of the orthogonality of Legendre kernel, we propose an alternative which results in sparse matrices and less computational cost.

The alternative idea is to apply the orthogonality of the kernel to change the minimization problem i.e. $v_M = \varphi_i$ in (4.10) as follows:

$$\begin{aligned} \min_{\mathbf{w},\varepsilon} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{\gamma}{2}\varepsilon^T\varepsilon \\ \text{s.t.} \quad & a(u_M^{n+1}, \varphi_i) - \langle F^n, \varphi_i\rangle = \varepsilon_i, \quad i = 0, 1, \cdots, M. \end{aligned} \tag{4.12}$$

This is Galerkin LS-SVR for the fourth-order FPDE (1.1)-(1.3).

**Theorem 3.** *The solution of problem* (4.11) *can be obtained uniquely by the following linear system:*

$$\left(\Omega - \frac{1}{\gamma}I\right)\alpha = \mathbf{y}, \tag{4.13}$$

*where* $\Omega_{ki} := \sum_{j=0}^{n}\left\langle \mathcal{L}\varphi_j(x), \tilde{\varphi}_i(x)\right\rangle\left\langle \mathcal{L}\varphi_j(x), \tilde{\varphi}_k(x)\right\rangle$ *for* $i, j = 0, \cdots, M$, $\mathbf{y} = \left[\langle F^n, \tilde{\varphi}_0(x)\rangle, \cdots, \langle F^n, \tilde{\varphi}_M(x)\rangle\right]^T$ *and* $\alpha = [\alpha_0, \cdots, \alpha_M]^T$ *are unknown Lagrange multipliers.*

*Proof.* Introducing Lagrange multipliers $\alpha_i, i = 0, 1, \cdots, M$, the Lagrangian function of problem (4.11) written as:

$$L(\mathbf{w}, \varepsilon, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{\gamma}{2}\varepsilon^T\varepsilon - \sum_{i=0}^{M}\alpha_i\left[\left\langle \mathcal{L}u^{n+1}(x), \tilde{\varphi}_i(x)\right\rangle - \langle F^n(x), \tilde{\varphi}_i(x)\rangle - \varepsilon_i\right].$$

Setting the partial derivatives equal to zero, the following sets of equations are obtained:

$$\frac{\partial L}{\partial w_k} = 0 \Rightarrow w_k = \sum_{i=0}^{M}\alpha_k\left\langle \mathcal{L}\varphi_k(x), \tilde{\varphi}_i(x)\right\rangle \qquad k = 0, 1, \ldots, M \tag{4.14}$$

$$\frac{\partial L}{\partial \varepsilon_k} = 0 \Rightarrow \gamma\varepsilon_k - \alpha_k = 0 \Rightarrow \varepsilon_k = \frac{1}{\gamma}\alpha_k \qquad k = 0, 1, \ldots, M \tag{4.15}$$

$$\frac{\partial L}{\partial \alpha_k} = 0 \Rightarrow \left\langle \mathcal{L}u^{n+1}(x), \tilde{\varphi}_k(x) \right\rangle - \left\langle F^n(x), \tilde{\varphi}_k(x) \right\rangle - \varepsilon_k = 0 \qquad k = 0, 1, \ldots, M \tag{4.16}$$

By using (4.5) and (4.15) in $\frac{\partial L}{\partial \alpha_k} = 0$, we get:

$$\sum_{j=0}^{n} w_j \left\langle \mathcal{L}\varphi_j(x), \tilde{\varphi}_k(x) \right\rangle - \frac{1}{\gamma}\alpha_k = \left\langle F^n(x), \tilde{\varphi}_k(x) \right\rangle \quad k = 0, 1, \ldots, M. \tag{4.17}$$

Now if we replace the weights $w_j$ from (4.14) in Eq (4.17), we have:

$$\sum_{i=0}^{M} \left( \sum_{j=0}^{n} \langle \mathcal{L}\varphi_j(x), \tilde{\varphi}_i(x) \rangle \langle \mathcal{L}\varphi_j(x), \tilde{\varphi}_k(x) \rangle \right) \alpha_i - \frac{1}{\gamma}\alpha_k = \langle F^n(x), \tilde{\varphi}_k(x) \rangle \quad k = 0, 1, \ldots, M. \tag{4.18}$$

We assume $\Omega_{ki} := \sum_{j=0}^{n} \left\langle \mathcal{L}\varphi_j(x), \tilde{\varphi}_i(x) \right\rangle \left\langle \mathcal{L}\varphi_j(x), \tilde{\varphi}_k(x) \right\rangle$ in (4.18), so we get:

$$\sum_{i=0}^{M} \Omega_{ki}\alpha_i - \frac{1}{\gamma}\alpha_k = \langle F^n(x), \tilde{\varphi}_k(x) \rangle \quad \text{for } k = 0, 1, \ldots, M$$

or equivalently,

$$\sum_{j=0}^{M} \Omega_{ij}\alpha_j - \frac{1}{\gamma}\alpha_i = \langle F^n(x), \tilde{\varphi}_i(x) \rangle \quad \text{for } i = 0, 1, \ldots, M, \tag{4.19}$$

which gives

$$u_M^{n+1} = \sum_{i=1}^{M} \alpha_i K(x, x_i), \tag{4.20}$$

as the appriximation solution at $t = t_{n+1}$. By solving the system of equations (4.19) we find $\alpha$ and then by (4.14) we can calculate unknown coefficients $w_j$ at each time step. The matrices present in the structure $\Omega$ are sparse matrices due to the utilization of orthogonal polynomials. □

*Remark* 4. It is worth noting that the coefficient matrix $(\Omega - \frac{1}{\gamma}I)$ remains constant for all time steps of the proposed algorithm, so a single efficient decomposition makes it available for all $t_n$.

*Remark* 5. Choosing the polynomial kernel in (4.20) in terms of shifted Legendre polynomials in the modal form (4.6), leads to well-conditioned matrices as illustrated in Figure 2. This figure also depicts a least-squares curve fitting with a second-degree polynomial and verifies the low-degree polynomial behaviour of the conditioning of the matrix. For example, for $\alpha = 0.5$ and $\eta = 1$ we get the second-degree polynomial $34.12x^2 - 4.03 \times 10^2 + 1.39 \times 10^3$. Moreover, the matrix is well-structured i.e., displaying a banded pattern as depicted in Figure 3. Additionally, The matrix is symmetric, positive definite, and diagonal dominant.

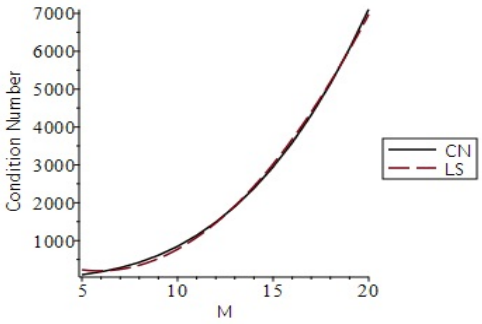


FIGURE 2. The condition number($CN$) for the coefficient matrix and a comparison with a polynomial least squares($LS$) fitting.

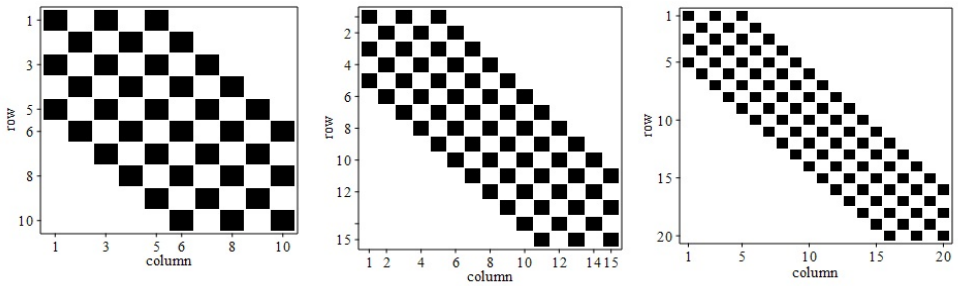


FIGURE 3. Band structure of the coefficient matrix for some numbers of basis functions $M = 10, 15, 20$, respectively.

## 5. NUMERICAL EXPERIMENTS

In this section, we provide some numerical simulation of the proposed method for the fourth-order FPDE (1.1)-(1.3). The errors at the desired final time $t = T$ are computed approximately as

$$L^2 \simeq \sqrt{\sum_{i=1}^{\mathcal{N}} \mid u(x_i,t_N) - u^N(x_i) \mid^2 \frac{1}{\mathcal{N}}} \tag{5.1}$$

where $u(x,t)$ and $u^N(x)$ are the exact and approximated solution respectivly. We set $\mathcal{N} = 100$ in all numerical results. In addition to the runtime and convergence of the proposed method, a measure of the order of convergence is also included in the numerical results. It is computed approximately as

$$EOC_i = \frac{\ln(e_{i+1}/e_i)}{\ln(M_i/M_{i+1})} \tag{5.2}$$

where $e_i$ given by (5.1) corresponding to the $M = M_i$. The runtime required for each iteration is likewise tabulated to characterize the computational efficiency, convergence, and practical applicability of our approach.

The computational runtimes reported correspond to the CPU time in seconds required to execute $M_i, i = 1,2,\cdots$ time steps to reach the final solution at the final time $t = T$. The numerical algorithms were implemented in Maple 2022 and executed on a laptop with a Core i7 processor and 8GB of RAM. We also set $\eta = 1$.

*Example* 1. As the first example, we consider the problem (1.1)-(1.3) with $f(x,t)$ associated to the exact solution $u(x) = t^{3+\alpha}\sin(2\pi x)$ as given in [16]. The goal is to check the convergence behavior of the proposed method both in spatial and temporal domains with different numbers of training data and several $\alpha$. In our investigation of spatial convergence, we have chosen the value of $N = 200$. Table 1 reports the $L^2$ errors (5.1) and runtimes for fractional orders $\alpha = 0.1, 0.25, 0.5, 0.75, 0.9$. Note that, $\alpha$ near 0 and 1 are also added to see how the performance of the method is affected by limiting values of $\alpha$. The experimental order of convergence (EOC) (5.2), is also reported in this table. The spatial convergence of the proposed scheme is reported in Table 2. The results show the exponential convergence of the proposed Crank-Nicolson and LS-SVR. This type of convergence for different values of $\alpha$ is well illustrated in Figure 4. in which The logarithmic scale for the error is linear i.e. the error is of exponential convergence.

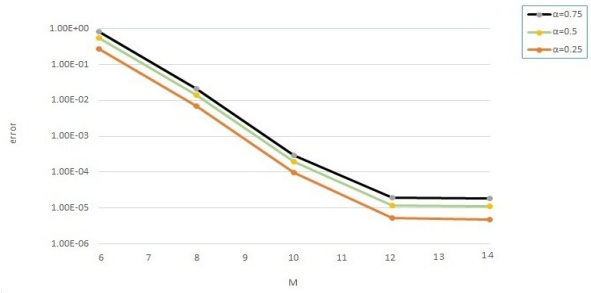


FIGURE 4. The spectral convergence of the proposed method for some fractional orders

*Example* 2. As the second example, we consider the problem (1.1)-(1.3) with $f(x,t)$ and the exact solution as $u = (t^{\alpha+2}+1)\cos(2\pi x)$ as given in [41]. We set $N = 100$ when investigating the convergence in space. Note that $N$ must be large enough to be able to ignore the effects of temporal errors on spatial convergence. Table 3 reports the $L^2$ errors (5.1) and runtimes for fractional orders $\alpha = 0.1, 0.25, 0.5, 0.75, 0.9$ and The spatial convergence of the proposed scheme is reported in Table 4. Also, convergence for different values of $\alpha$ is well illustrated in Figure 5.

TABLE 1. The error and temporal order of convergence

| | $\alpha = 0.1$ | | $\alpha = 0.25$ | | $\alpha = 0.5$ | | $\alpha = 0.75$ | | $\alpha = 0.9$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| $M$ | Error | EOC | Error | EOC | Error | EOC | Error | EOC | Error | EOC |
| 6 | 2.77E-01 | | 2.77E-01 | | 2.77E-01 | | 2.77E-01 | | 2.77E-01 | |
| 8 | 7.06E-03 | 12.76 | 7.06E-03 | 12.76 | 7.06E-03 | 12.76 | 7.06E-03 | 12.76 | 7.06E-03 | 12.76 |
| 10 | 9.65E-05 | 19.24 | 9.67E-05 | 19.23 | 9.72E-05 | 19.20 | 9.75E-05 | 19.19 | 9.69E-05 | 19.22 |
| 12 | 4.35E-06 | 17.00 | 5.12E-06 | 16.12 | 6.70E-06 | 14.68 | 7.62E-06 | 13.98 | 5.54E-06 | 15.69 |
| 14 | 3.84E-06 | 0.81 | 4.64E-06 | 0.64 | 6.25E-06 | 0.44 | 7.19E-06 | 0.37 | 5.07E-06 | 0.57 |

TABLE 2. The spatial convergence of the proposed scheme for $M = 16$

| | $\alpha = 0.25$ | | $\alpha = 0.5$ | | $\alpha = 0.75$ | |
|---|---|---|---|---|---|---|
| $N$ | Error | EOC | Error | EOC | Error | EOC |
| 40 | 2.29E-05 | | 2.95E-05 | | 3.07E-05 | |
| 80 | 1.16E-05 | 0.99 | 1.52E-05 | 0.95 | 1.66E-05 | 0.88 |
| 160 | 5.81E-06 | 0.99 | 7.79E-06 | 0.97 | 8.85E-06 | 0.91 |
| 320 | 2.92E-06 | 0.99 | 3.96E-06 | 0.98 | 4.65E-06 | 0.93 |
| 640 | 1.46E-06 | 1.00 | 2.00E-06 | 0.98 | 2.42E-06 | 0.94 |

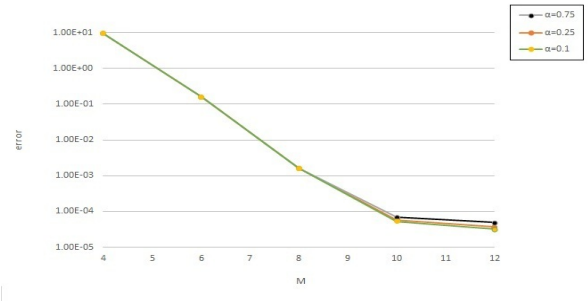


FIGURE 5. The spectral convergence of the proposed method for some fractional orders

TABLE 3. The error, temporal order of convergence, and CPU time in seconds

| | | $\alpha = 0.1$ | | $\alpha = 0.25$ | | $\alpha = 0.5$ | | $\alpha = 0.75$ | | $\alpha = 0.9$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $M$ | time | Error | EOC | Error | EOC | Error | EOC | Error | EOC | Error | EOC |
| 4 | 16.31 | 9.45E+00 | | 9.45E+00 | | 9.44E+00 | | 9.43E+00 | | 9.42E+00 | |
| 6 | 28.44 | 1.60E-01 | 10.06 | 1.60E-01 | 10.06 | 1.60E-01 | 10.06 | 1.60E-01 | 10.06 | 1.59E-01 | 10.06 |
| 8 | 48.38 | 1.61E-03 | 15.98 | 1.61E-03 | 15.98 | 1.61E-03 | 15.97 | 1.61E-03 | 15.97 | 1.60E-03 | 15.97 |
| 10 | 75.61 | 5.29E-05 | 15.31 | 5.75E-05 | 14.94 | 6.62E-05 | 14.31 | 6.90E-05 | 14.13 | 5.36E-05 | 15.25 |
| 12 | 115.47 | 3.21E-05 | 2.74 | 3.71E-05 | 2.40 | 4.65E-05 | 1.94 | 4.95E-05 | 1.82E | 3.29E-05 | 2.67 |

TABLE 4. The spatial convergence of the proposed scheme for $M = 16$

| | $\alpha = 0.25$ | | $\alpha = 0.5$ | | $\alpha = 0.75$ | |
|---|---|---|---|---|---|---|
| N | Error | EOC | Error | EOC | Error | EOC |
| 40 | 9.46E-05 | | 1.15E-04 | | 1.16E-04 | |
| 80 | 4.76E-05 | 0.99 | 5.90E-05 | 0.96 | 6.20E-05 | 0.90 |
| 160 | 2.39E-05 | 0.99 | 3.00E-05 | 0.98 | 3.27E-05 | 0.92 |
| 320 | 1.20E-05 | 1.00 | 1.52E-05 | 0.98 | 1.71E-05 | 0.94 |
| 640 | 5.99E-06 | 1.00 | 7.67E-06 | 0.99 | 8.83E-06 | 0.95 |
| 1280 | 3.00E-06 | 1.00 | 3.86E-06 | 0.99 | 4.54E-06 | 0.96 |

## 6. CONCLUSION

In this work, we propose a supervised ML algorithm, least squares support vector regression for the fractional fourth-order PDEs. To enhance computational efficiency, we employ orthogonal polynomial kernels. Through various test problems, we demonstrate that our proposed approach achieves exponential convergence.

## REFERENCES

[1] P. Ahadian and K. Parand, "Support vector regression for the temperature-stimulated drug release," *Chaos, Solitons & Fractals*, vol. 165, p. 112871, 2022, doi: 10.1016/j.chaos.2022.112871.

[2] S. Arabi, A. El-Said, and M. Laghrouche, "Numerical solutions of fractional fourth order partial differential equations using finite difference method," *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 6263–6273, 2011.

[3] T. Barth and M. Ohlberger, *Finite volume methods: foundation and analysis*. John Wiley & Sons, 2017. doi: 10.1002/9781119176817.ecm2010.

[4] M. Caputo and M. Fabrizio, "A new definition of fractional derivative without singular kernel," *Progress in Fractional Differentiation & Applications*, vol. 1, no. 2, pp. 73–85, 2015, doi: 10.12785/pfda/010201.

[5] G. Chai, "Positive solutions for boundary value problem of fractional differential equation with p-laplacian operator," *Boundary Value Problems*, vol. 2012, pp. 1–20, 2012, doi: 10.1186/1687-2770-2012-18.

[6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995, doi: 10.1007/BF00994018.

[7] Y. Dibike, S. Velickov, and D. Solomatine, "Support vector machines: Review and applications in civil engineering," in *Proceedings of the 2nd Joint Workshop on Application of AI in Civil Engineering*, Mar 26 2000, pp. 215–218.

[8] B. Duan, B. Jin, R. Lazarov, J. Pasciak, and Z. Zhou, "Space-time petrov-galerkin fem for fractional diffusion problems," *Computational Methods in Applied Mathematics*, vol. 18, no. 1, pp. 1–20, Jan 1 2018, doi: 10.1515/cmam-2017-0026.

[9] R. Eymard, T. Gallouet, and R. Herbin, *Finite volume methods*, ser. Handbook of Numerical Analysis. Elsevier, 2000, vol. 7, doi: 10.1016/S1570-8659(00)07005-8.

[10] J. Gracia, E. O'Riordan, and M. Stynes, "Convergence in positive time for a finite difference method applied to a fractional convection-diffusion problem," *Computational Methods in Applied Mathematics*, vol. 18, no. 1, pp. 33–42, 2018.

[11] S. Guo, L. Mei, Z. Zhang, J. Chen, Y. He, and Y. Li, "Finite difference/hermite–galerkin spectral method for multi-dimensional time-fractional nonlinear reaction–diffusion equation in unbounded domains," *Applied Mathematical Modelling*, vol. 70, pp. 246–263, 2019, doi: 10.1016/j.apm.2019.01.018.

[12] X. Guo, C. Wu, M. Marchese, and Y. Liang, "Ls-svr-based solving volterra integral equations," *Applied Mathematics and Computation*, vol. 218, no. 23, pp. 11 404–11 409, 2012, doi: 10.1016/j.amc.2012.05.028.

[13] N. S. Hamzehkolaei and N. Kadkhoda, "An efficient ranked voronoi diagram-based hybrid method for reliability-based structural analysis and design optimization," *Soft Comput.*, vol. 27, no. 19, p. 13889–13910, may 2023, doi: 10.1007/s00500-023-08450-z.

[14] H. Jafari and N. Kadkhoda, "Application of simplest equation method to the (2+ 1)-dimensional nonlinear evolution equations," *New Trends in Mathematical Sciences*, vol. 2, no. 2, pp. 64–68, 2014.

[15] A. Jafarian and S. M. Nia, "Utilizing feed-back neural network approach for solving linear fredholm integral equations system," *Applied Mathematical Modelling*, vol. 37, no. 7, pp. 5027–5038, 2013, doi: 10.1016/j.apm.2012.09.029.

[16] M. Jani, E. Babolian, and D. Bhatta, "A petrov–galerkin spectral method for the numerical simulation and analysis of fractional anomalous diffusion," *Mathematical Methods in the Applied Sciences*, vol. 44, no. 2, pp. 2021–2032, 2021, doi: 10.1002/mma.6908.

[17] S. Javadi, M. Jani, and E. Babolian, "A numerical scheme for space-time fractional advection-dispersion equation," *arXiv preprint arXiv:1512.06629*, 2015, doi: 10.48550/arXiv.1512.06629.

[18] Y. Jiang and J. Ma, "High-order finite element methods for time-fractional partial differential equations," *Journal of Computational and Applied Mathematics*, vol. 235, no. 11, pp. 3285–3290, 2011, doi: 10.1016/j.cam.2011.01.011.

[19] B. Jin, R. Lazarov, and Z. Zhou, "Two fully discrete schemes for fractional diffusion and diffusion-wave equations with nonsmooth data," *SIAM journal on scientific computing*, vol. 38, no. 1, pp. A146–A170, 2016, doi: 10.1137/140979563.

[20] N. Kadkhoda, "Application of $\frac{G'}{G^2}$-expansion method for solving fractional differential equations," *International journal of applied and computational mathematics*, vol. 3, pp. 1415–1424, 2017, doi: 10.1007/s40819-017-0344-2.

[21] A. Kufner, O. John, and S. Fucik, *Function spaces*. Springer Science & Business Media, 1977, vol. 3.

[22] S. Kumar, "Numerical solution of fuzzy fractional diffusion equation by chebyshev spectral method," *Numerical Methods for Partial Differential Equations*, vol. 38, no. 3, pp. 490–508, 2022, doi: 10.1002/num.22650.

[23] Y. Lin and C. Xu, "Finite difference/spectral approximations for the time-fractional diffusion equation," *Journal of computational physics*, vol. 225, no. 2, pp. 1533–1552, 2007, doi: 10.1016/j.jcp.2007.02.001.

[24] Y. Liu and J. Wu, "Fixed point theorems in piecewise continuous function spaces and applications to some nonlinear problems," *Mathematical Methods in the Applied Sciences*, vol. 37, no. 4, pp. 508–517, 2014, doi: 10.1002/mma.2809.

[25] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "Deepxde: A deep learning library for solving differential equations," *SIAM review*, vol. 63, no. 1, pp. 208–228, 2021, doi: 10.1137/19M1274067.

[26] S. Mall and S. Chakraverty, "Chebyshev neural network based model for solving lane–emden type equations," *Applied Mathematics and Computation*, vol. 247, pp. 100–114, 2014, doi: 10.1016/j.amc.2014.08.085.

[27] S. Mehrkanoon, T. Falck, and J. A. Suykens, "Approximate solutions to ordinary differential equations using least squares support vector machines," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 9, pp. 1356–1367, 2012, doi: 10.1109/TNNLS.2012.2202126.

[28] S. Mehrkanoon and J. A. Suykens, "Learning solutions to partial differential equations using ls-svm," *Neurocomputing*, vol. 159, pp. 105–116, 2015, doi: 10.1016/j.neucom.2015.02.013.
[29] R. Metzler and J. Klafter, "The random walk's guide to anomalous diffusion: a fractional dynamics approach," *Physics reports*, vol. 339, no. 1, pp. 1–77, 2000, doi: 10.1016/S0370-1573(00)00070-3.
[30] F. Moukalled, L. Mangani, M. Darwish, F. Moukalled, L. Mangani, and M. Darwish, *The finite volume method*. Springer, 2016.
[31] O. Nikan, J. T. Machado, and A. Golbabai, "Numerical solution of time-fractional fourth-order reaction-diffusion model arising in composite environments," *Applied Mathematical Modelling*, vol. 89, pp. 819–836, 2021, doi: 10.1016/j.apm.2020.07.021.
[32] K. Parand, A. A. Aghaei, M. Jani, and A. Ghodsi, "A new approach to the numerical solution of fredholm integral equations using least squares-support vector regression," *Mathematics and Computers in Simulation*, vol. 180, pp. 114–128, 2021, doi: 10.1016/j.matcom.2020.08.010.
[33] K. Parand, A. Aghaei, M. Jani, and A. Ghodsi, "Parallel ls-svm for the numerical simulation of fractional volterra's population model," *Alexandria Engineering Journal*, vol. 60, no. 6, pp. 5637–5647, 2021, doi: 10.1016/j.aej.2021.04.034.
[34] I. Podlubny, *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*. Elsevier, 1998.
[35] P. Rahimkhani, Y. Ordokhani, and S. Sabermahani, "Bernoulli wavelet least squares support vector regression: Robust numerical method for systems of fractional differential equations," *Mathematical Methods in the Applied Sciences*, vol. 46, no. 17, pp. 17 641–17 659, 2023, doi: 10.1002/mma.9522.
[36] R. Scherer, S. L. Kalla, Y. Tang, and J. Huang, "The grünwald–letnikov method for fractional differential equations," *Computers & Mathematics with Applications*, vol. 62, no. 3, pp. 902–917, 2011, doi: 10.1016/j.camwa.2011.03.054.
[37] J. Shen, T. Tang, and L.-L. Wang, *Spectral methods: algorithms, analysis and applications*. Springer Science & Business Media, 2011, vol. 41.
[38] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, pp. 199–222, Aug 2004, doi: 10.1023/B:STCO.0000035301.49549.88.
[39] J. A. Suykens and J. Vandewalle, "Recurrent least squares support vector machines," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 7, pp. 1109–1114, 2000, doi: 10.1109/81.855471.
[40] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.
[41] Y. Zhu and Z.-Z. Sun, "A high-order difference scheme for the space and time fractional bloch–torrey equation," *Computational Methods in Applied Mathematics*, vol. 18, no. 1, pp. 147–164, 2018, doi: 10.1515/cmam-2017-0034.

*Author's address*

**Marziyeh Felahat**
Bozorgmehr University Of Qaenat, Department Of Basic Sciences, Qaenat, Iran
*E-mail address:* m_felahat@buqaen.ac.ir