Analytical and Numerical Solutions for Nonlinear Equations
©Available online at https://ansne.du.ac.ir/
Online ISSN: 3060–785X
2024, Volume 9, Issue 2, pp. 193–202
Research article

A Physics-Informed LSSVR Method with Legendre Kernels for Direct Solution of Fokker-Planck Equations

Maryam Babaei[1], Kourosh Parand[1,2,*]

[1] Department of Computer and Data Sciences, Faculty of Mathematical Sciences, Shahid Beheshti University, Tehran, Iran

[2] Institute for Cognitive and Brain Sciences, Shahid Beheshti University, Tehran, Iran

*** Corresponding author(s):** k_parand@sbu.ac.ir

Received: 07/09/2025 **Revised:** 17/09/2025 **Accepted:** 22/09/2025 **Published:** 27/09/2025 10.22128/ansne.2025.3047.1151

Abstract

The FokkerPlanck equation is widely used to describe how systems evolve when randomness plays a role. It appears in many fields, including physics, finance, biology, and engineering. Classical numerical methods usually require discretization, which can make the computation expensive, unstable, or less accurate. In this work, we present a direct method for solving these equations using Least Squares Support Vector Regression (LSSVR) with Legendre kernels. Our approach avoids discretization and provides global optimization, which helps overcome the difficulties faced by loss-based methods such as Physics-Informed Neural Networks (PINNs). The use of Legendre kernels gives strong approximation properties and ensures high accuracy in the solutions. We tested the method on several problems and found that it achieves very precise results while being faster and more stable than PINNs. To further improve reliability, we also applied automatic hyperparameter tuning, which adapts the method to each problem without manual adjustment. These results show that LSSVR with Legendre kernels is a simple, accurate, and efficient tool for scientists and engineers who need to solve FokkerPlanck equations.

Keywords: Fokker-Planck equation, Least squares support vector regression, Legendre kernels, Physics-informed machine learning, Kernel methods

Mathematics Subject Classification (2020): 35Q84, 65M70, 68T07, 33C45

1 Introduction

The Fokker-Planck equation is one of the most important partial differential equations in statistical physics and stochastic processes. It describes how probability density functions change over time for systems with random fluctuations [1,2]. It has been widely used in many fields, including financial modeling [3], population dynamics [4], quantum mechanics [5], and engineering systems [6].

The general form of the linear Fokker-Planck equation can be written as:

$$\frac{\partial p(x,t)}{\partial t} = -\frac{\partial}{\partial x}[A(x,t)p(x,t)] + \frac{1}{2}\frac{\partial^2}{\partial x^2}[B(x,t)p(x,t)], \tag{1}$$

where $p(x,t)$ is the probability density function, $A(x,t)$ is the drift coefficient, and $B(x,t)$ is the diffusion coefficient. For linear cases, these coefficients depend only on space and time, not on the probability density function itself [7,8].

Traditional numerical methods for solving Fokker-Planck equations include finite difference methods [9], finite element methods [10], spectral methods [11], and Monte Carlo approaches [12]. However, these methods face several challenges: they require spatial and temporal discretization, which increases computational complexity; they can be unstable for certain parameter ranges; and they may not provide sufficient accuracy for complex problems [13].

Recent advances in machine learning and kernel-based methods have opened new ways to solve partial differential equations. Least Squares Support Vector Regression (LSSVR) has emerged as a powerful tool for solving ODEs and PDEs due to its strong theoretical foundations and computational efficiency [14, 15]. LSSVR transforms the PDE problem into a convex optimization problem, offering advantages in finding global solutions and avoiding local minima common in neural network-based methods. Several studies have successfully applied LSSVR for various PDE problems, demonstrating high accuracy at relatively low computational cost [16]. Recent developments in kernel design have further enhanced the performance of kernel-based methods. Behdani and Darehmiraki demonstrated that Wendland functions with compact support can significantly improve kernel ridge regression models, offering computational advantages through sparsity while maintaining high approximation accuracy [17]. Notably, Rad et al. introduced fractional orthogonal kernel classifiers in support vector machines, highlighting the effectiveness of orthogonal kernels for solving integral and differential equations [18]. Similarly, optimization-based metaheuristic approaches have been applied for solving differential and integral equations efficiently [19–22].

Physics-Informed Neural Networks (PINNs) have also gained attention for PDE solving [23, 24]. However, PINNs face several challenges: they rely on loss function minimization, which can get trapped in local minima; require careful tuning of loss function weights; and can be slow and computationally expensive to train [25]. These limitations make PINNs less reliable for applications requiring fast and accurate solutions [26, 27].

Legendre polynomials and their associated kernels are valuable in numerical analysis due to their excellent approximation properties and orthogonality [28, 29]. They form a complete orthogonal basis in the space of square-integrable functions, making them suitable for spectral and kernel-based methods [30]. Recent studies have demonstrated the effectiveness of Legendre kernels in machine learning and differential equation solvers [31–33]. Complementary to Legendre-based approaches, pseudo-spectral methods using Lagrange polynomials have shown promise for solving fractional differential equations, as demonstrated by Mirzaei and Shokri in their work on fractional cable equations [34].

The motivation for developing the proposed LSSVRLegendre framework is to provide a PDE solver that combines the accuracy of spectral methods, the stability of kernel-based approaches, and the efficiency of convex optimization. Unlike classical discretization methods, which suffer from grid-induced errors and high computational cost, our approach represents the solution globally without spatialtemporal meshing. Compared to PINNs, which rely on non-convex loss minimization and often face convergence and hyperparameter tuning challenges, LSSVR guarantees a unique global solution through convex optimization. Incorporating Legendre kernels further enhances numerical stability and approximation accuracy due to their orthogonality and spectral-like convergence properties, making the framework robust and efficient for solving FokkerPlanck equations and related PDEs across scientific and engineering domains.

Although numerical methods have progressed, critical challenges remain. Traditional methods require domain discretization, increasing computational cost and introducing numerical errors. Ensuring numerical stability and convergence is difficult, and loss-based methods like neural networks can get trapped in local minima, compromising solution quality. Current methods often force a trade-off between computational efficiency and solution accuracy [35].

Addressing these challenges is important across multiple scientific domains. In financial modeling, accurate solutions to Fokker-Planck equations are crucial for option pricing and risk assessment [3]. In biological systems, these equations model population dynamics and evolutionary processes, where traditional methods may be insufficient [4, 36]. In engineering, understanding stochastic system behavior requires robust numerical tools [6].

This research investigates whether LSSVR with Legendre kernels can provide a more efficient and accurate approach than traditional discretization and neural network methods. Our method solves the PDE directly without spatial-temporal grid discretization, converges faster than PINNs, avoids local minima, and leverages Legendre kernels for high accuracy. Hyperparameter optimization ensures optimal performance across different problem types.

The remainder of this paper is organized as follows. Section 2 presents the methodology, including the LSSVR framework, Legendre kernel formulation, and algorithm. Section 3 provides numerical results and comparisons to existing methods. Finally, Section 4 concludes with a discussion of findings and future research directions.

2 Methodology

In this section, we present the theoretical foundation and algorithmic framework for solving Fokker-Planck equations using least squares support vector regression with Legendre kernels.

2.1 Least Squares Support Vector Regression for PDEs

LSSVR is a powerful machine learning technique that can be adapted for solving differential equations. For a linear Fokker-Planck equation, we seek an approximate solution in the form:

$$u(x,t) = \sum_{i=1}^{N_{\text{int}}} \alpha_i K((x_i,t_j),(x,t)) + b. \tag{2}$$

Here, $K((x_i,t_j),(x,t))$ is the kernel function, α_i are the support vector coefficients, and b is the bias term.

The LSSVR formulation for PDE solving involves minimizing the following objective function:

$$J(\alpha,b,e) = \frac{1}{2}\sum_{i=1}^{N_{\text{int}}} \alpha_i^2 + \frac{\gamma}{2}\sum_{i=1}^{N_{\text{int}}} e_i^2. \tag{3}$$

This is subject to the equality constraints:

$$\mathscr{L}\left[\sum_{j=1}^{N_{\text{int}}} \alpha_j K((x_j,t_j),(x_i,t_i)) + b\right] = f(x_i,t_j) + e_i, \quad i = 1,2,\ldots,N_{\text{int}}. \tag{4}$$

Here, $\mathscr{L}$ is the linear differential operator for the Fokker-Planck equation, $f(x_i,t_j)$ represents the known right-hand side terms, and e_i are the error variables.

Using the method of Lagrange multipliers, we introduce multipliers λ_i and form the Lagrangian:

$$\begin{aligned}\mathscr{L}(\alpha,b,e,\lambda) = {} & \frac{1}{2}\sum_{i=1}^{N_{\text{int}}} \alpha_i^2 + \frac{\gamma}{2}\sum_{i=1}^{N_{\text{int}}} e_i^2 \\ & - \sum_{i=1}^{N_{\text{int}}} \lambda_i \left[\mathscr{L}\left(\sum_{j=1}^{N_{\text{int}}} \alpha_j K((x_j,t_j),(x_i,t_i)) + b\right) - f(x_i,t_j) - e_i\right].\end{aligned} \tag{5}$$

Applying the Karush-Kuhn-Tucker (KKT) optimality conditions:

$$\frac{\partial \mathscr{L}}{\partial \alpha_k} = 0 \;\Rightarrow\; \alpha_k = \sum_{i=1}^{N_{\text{int}}} \lambda_i \mathscr{L}[K((x_k,t_k),(x_i,t_j))], \quad k = 1,\ldots,N_{\text{int}}, \tag{6}$$

$$\frac{\partial \mathscr{L}}{\partial b} = 0 \;\Rightarrow\; \sum_{i=1}^{N_{\text{int}}} \lambda_i = 0, \tag{7}$$

$$\frac{\partial \mathscr{L}}{\partial e_i} = 0 \;\Rightarrow\; \gamma e_i = \lambda_i, \quad i = 1,\ldots,N_{\text{int}}, \tag{8}$$

$$\frac{\partial \mathscr{L}}{\partial \lambda_i} = 0 \;\Rightarrow\; \mathscr{L}\left[\sum_{j=1}^{N_{\text{int}}} \alpha_j K((x_j,t_j),(x_i,t_j)) + b\right] = f(x_i,t_j) + e_i. \tag{9}$$

The dual formulation leads to the linear system:

$$\begin{bmatrix} \Omega + \gamma^{-1}\mathbf{I} & \mathbf{K}_{\text{bnd}}^T \\ \mathbf{K}_{\text{bnd}} & \mathbf{K}_{\text{bnd,bnd}} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{y}_{\text{bnd}} \end{bmatrix}. \tag{10}$$

Here, $\Omega_{ij} = \mathscr{L}\left[\mathscr{L}\left[K((x_i,t_i),(x_j,t_j))\right]\right]$ is the operator matrix for interior points, $\mathbf{K}_{\text{bnd}}$ is kernel evaluations between boundary and interior points, and $\mathbf{K}_{\text{bnd,bnd}}$ shows kernel evaluations between boundary points. The vector $\mathbf{y}_{\text{bnd}}$ contains boundary and initial condition values, while α are coefficients for interior points and β are coefficients for boundary points.

2.2 Legendre Kernels

Legendre polynomials provide an excellent basis for approximating smooth functions due to their orthogonality properties and convergence characteristics. The Legendre kernel is constructed using the generating function of Legendre polynomials:

$$K_L(x,y) = \sum_{n=0}^{\infty} \frac{P_n(x)P_n(y)}{2n+1}. \tag{11}$$

Here, $P_n(x)$ are the Legendre polynomials of degree n, defined by Rodrigues' formula:

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} \left[(x^2-1)^n \right]. \tag{12}$$

For practical implementation, we use a truncated version:

$$K_L(x,y) = \sum_{n=0}^{M} \frac{P_n(x)P_n(y)}{2n+1}. \tag{13}$$

Here, M is the maximum degree of polynomials considered.

The Legendre kernel has several advantageous properties. Legendre polynomials satisfy orthogonality:

$$\int_{-1}^{1} P_m(x)P_n(x)\,dx = \frac{2}{2n+1}\delta_{mn}.$$

They provide uniform convergence for smooth functions and allow efficient computation of derivatives required for differential operators.

The derivatives of the Legendre kernel can be computed analytically:

$$\frac{\partial K_L(x,y)}{\partial x} = \sum_{n=0}^{M} \frac{P_n'(x)P_n(y)}{2n+1}, \tag{14}$$

$$\frac{\partial^2 K_L(x,y)}{\partial x^2} = \sum_{n=0}^{M} \frac{P_n''(x)P_n(y)}{2n+1}. \tag{15}$$

For two-dimensional problems with variables (x,t), we construct the tensor product kernel:

$$K_{2D}((x_i,t_j),(x_k,t_l)) = K_L^x(x_i,x_k) \cdot K_L^t(t_j,t_l), \tag{16}$$

where K_L^x and K_L^t are one-dimensional Legendre kernels with degrees M_x and M_t, respectively.

2.3 Algorithm for LSSVR with Legendre Kernels

The complete algorithm for solving linear Fokker-Planck equations combines LSSVR with Legendre kernels using the dual formulation. Algorithm 2.3 shows the main steps of our approach. [1] **Input:** Fokker-Planck equation, domain Ω, boundary conditions, initial conditions **Parameters:** Degrees M_x, M_t, regularization γ, collocation points n_x, n_t **Generate Collocation Points:** Interior points $\{(x_i,t_j)\}_{i=1,j=1}^{N_{x,\text{int}},N_{t,\text{int}}}$ Boundary points $\{(x_i,t_j)\}_{i=1,j=1}^{N_{x,\text{bnd}},N_{t,\text{bnd}}}$ for initial and boundary conditions **Construct Kernel Matrices:** Compute $\Omega_{(i,j),(k,l)} = \mathscr{L}[K_{2D}((x_i,t_j),(x_k,t_l))]$ for interior points Compute $\mathbf{K}_{\text{bnd,int}}$ between boundary and interior points Compute $\mathbf{K}_{\text{bnd,bnd}}$ between boundary points **Assemble System Matrix:** $\mathbf{M} = \begin{bmatrix} \Omega + \gamma^{-1}\mathbf{I} & \mathbf{K}_{\text{bnd,int}}^T \\ \mathbf{K}_{\text{bnd,int}} & \mathbf{K}_{\text{bnd,bnd}} \end{bmatrix}$ **Assemble Right-Hand Side:** $\mathbf{b} = \begin{bmatrix} \mathbf{0} \\ \mathbf{y}_{\text{bnd}} \end{bmatrix}$ where $\mathbf{y}_{\text{bnd}}$ contains boundary/initial values **Solve Linear System:** Solve $\mathbf{M}\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \mathbf{b}$ **Construct Solution:** $u(x,t) = \sum_{i,j} \alpha_{ij} K_{2D}((x,t),(x_i,t_j)) + \sum_{k,l} \beta_{kl} K_{2D}((x,t),(x_k,t_l))$ **Output:** Approximate solution $u(x,t)$ For optimal performance, we use hyperparameter optimization to determine the best values for polynomial degrees M_x and M_t, the regularization parameter γ, and the number of collocation points n_x and n_t. The optimization process uses the Optuna framework to minimize validation error across parameter space [37]. This ensures optimal accuracy for each specific problem type.

The computational complexity of the algorithm is $O(N^3)$ where $N = N_{\text{int}} + N_{\text{bnd}}$ is the total number of collocation points. This is due to the linear system solution. The method requires only a single solve without iterative procedures, making it computationally efficient compared to neural network approaches that require many training epochs.

3 Numerical Results

In this section, we present comprehensive numerical experiments to validate the effectiveness of our proposed LSSVR method with Legendre kernels for solving Fokker-Planck equations. To evaluate the accuracy of our numerical solutions, we use three standard error metrics:

- Root Mean Square (RMS) Error:

$$\text{RMS} = \sqrt{\frac{1}{N_x N_t}\sum_{i=1}^{N_x}\sum_{j=1}^{N_t}\left(u_{\text{exact}}(x_i,t_j) - u_{\text{approx}}(x_i,t_j)\right)^2} \tag{17}$$

- Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{N_x N_t}\sum_{i=1}^{N_x}\sum_{j=1}^{N_t}\left|u_{\text{exact}}(x_i,t_j) - u_{\text{approx}}(x_i,t_j)\right| \tag{18}$$

- Relative L_2 Error:

$$\text{Relative } L_2 = \frac{\sqrt{\sum_{i=1}^{N_x}\sum_{j=1}^{N_t}\left(u_{\text{exact}}(x_i,t_j) - u_{\text{approx}}(x_i,t_j)\right)^2}}{\sqrt{\sum_{i=1}^{N_x}\sum_{j=1}^{N_t}\left(u_{\text{exact}}(x_i,t_j)\right)^2}} \tag{19}$$

Here, N_x and N_t denote the total number of spatial and temporal test points, respectively.

Example 1. We consider the Fokker-Planck equation:

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial u(x,t)}{\partial x} + \frac{\partial^2 u(x,t)}{\partial x^2}, \quad (x,t) \in [0,1]\times[0,1]. \tag{20}$$

The initial condition is $u(x,0) = x$, and the boundary conditions are $u(0,t) = t$, $u(1,t) = 1+t$. The exact solution is $u(x,t) = x+t$ [38].

The optimal parameters obtained through Optuna optimization are $M_x = 9$, $M_t = 5$, $\gamma = 5\times 10^7$. We use collocation points $n_{\text{colloc}_x} = 19$ and $n_{\text{colloc}_t} = 30$. The total number of training points is 600. We used 8000 test points for validation.

The present method achieves exceptional accuracy for this linear problem, as shown in Table 1. This method outperforms both B-spline methods [38] and Physics-Informed Neural Networks (PINNs) [39]. Figure 1 shows the method solution compared to the exact solution.

Table 1. Error and timing comparison for example 1 using different methods

Method	Present Method		B-Spline [38]	PINN [39]
	Train	Test		
Relative L_2	1.379581e-11	8.284437e-12	$\sim 10^{-6}$*	1.0617e-04
RMS Error	1.528015e-11	8.962584e-12	-	1.1567e-04
MAE Error	8.942321e-12	5.136961e-12	-	1.0073e-04
Solver Time (s)	0.070		-	-
Test Eval Time (s)	0.499		-	-

*Estimated from absolute errors in Table I

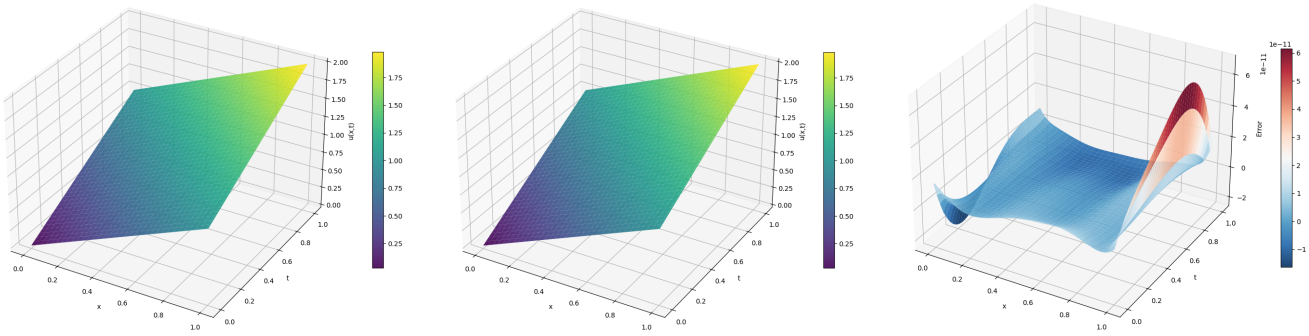
Example 2. Consider the Fokker-Planck equation:

$$\frac{\partial u(x,t)}{\partial t} = -\frac{\partial}{\partial x}\left[A(x,t)\,u(x,t)\right] + \frac{\partial^2}{\partial x^2}\left[B(x,t)\,u(x,t)\right], \tag{21}$$

where $A(x,t) = \frac{4}{x} - \frac{x}{3}$ and $B(x,t) = u(x,t)$. The exact solution is $u(x,t) = x^2 e^t$ [40].

The optimized parameters are $M_x = 3$, $M_t = 15$, $\gamma = 1\times 10^5$, tolerance = 1×10^{-10}. We use collocation points $n_{\text{colloc}_x} = 24$ and $n_{\text{colloc}_t} = 20$. The training set consists of 600 points. We used 8000 test points.

As presented in Table 2, the present method shows good performance compared to the Adomian Decomposition Method (ADM) [40] by approximately two orders of magnitude. It outperforms PINNs [39] by more than six orders of magnitude. The Solution and error distribution shown in Figure 2.

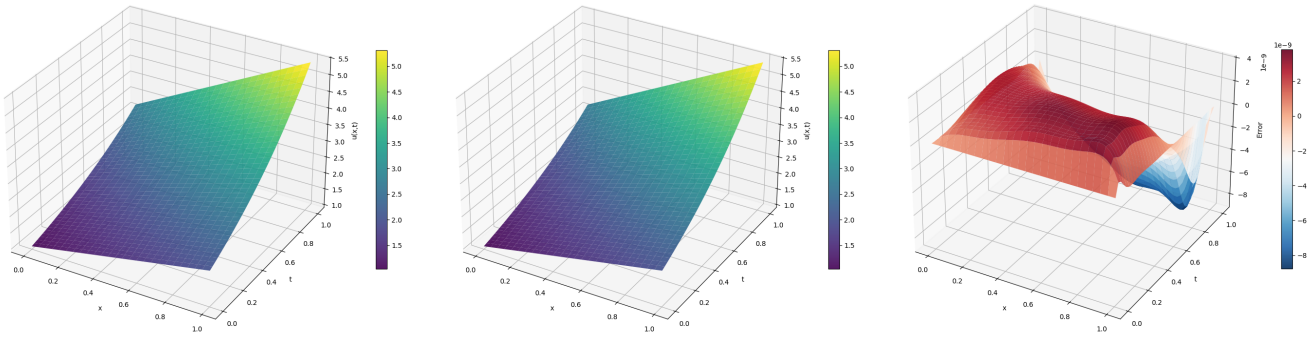


(a) LSSVR-Legendre solution (b) Exact solution (c) Error distribution

Figure 1. Comparison of solutions for Example 1. Subfigure (a) shows the approximate solution obtained using the proposed LSSVR-Legendre method, (b) presents the exact analytical solution, and (c) illustrates the error distribution across the domain.

Table 2. Error and timing comparison for example 2 using different methods

Method	Present Method		ADM [40]	PINN [39]
	Train	Test		
Relative L_2	3.171752e-10	3.092609e-10	5.61e-08	4.0720e-04
RMS Error	3.066744e-10	2.492252e-10	-	4.2857e-04
MAE Error	1.548611e-10	1.556033e-10	-	3.2648e-04
Solver Time (s)	0.635		-	-
Test Eval Time (s)	0.381		-	-



(a) LSSVR-Legendre solution (b) Exact solution (c) Error distribution

Figure 2. Comparison of solutions for Example 2. Subfigure (a) shows the numerical solution obtained with the proposed framework, (b) gives the exact solution, and (c) displays the error distribution.

Example 3 (Backward Kolmogorov Equation)**.** We consider the backward Kolmogorov equation:

$$\frac{\partial u(x,t)}{\partial t} = -A(x,t)\frac{\partial u(x,t)}{\partial x} + B(x,t)\frac{\partial^2 u(x,t)}{\partial x^2}, \tag{22}$$

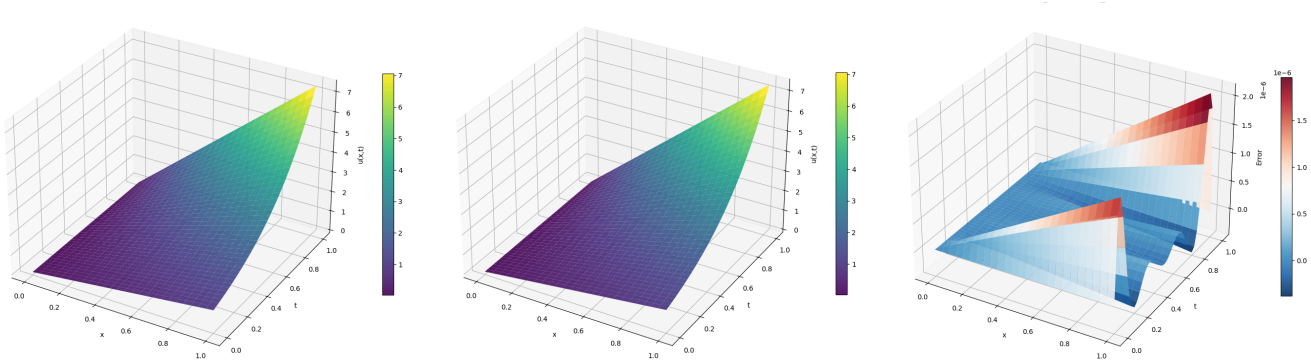
where $A(x,t) = -(x+1)$ and $B(x,t) = x^2 e^t$. The exact solution is $u(x,t) = (x+1)e^t$ [41].

The optimal parameters are $M_x = 8$, $M_t = 12$, $\gamma = 1 \times 10^6$. We use collocation points $n_{\text{colloc}_x} = 19$ and $n_{\text{colloc}_t} = 21$. The training set contains 600 points and 8000 test points.

As presented in Table 3, the present method achieves significantly better accuracy than both ADM [40] and Hermite Radial Basis Functions (HRBF) [41] methods. The solution quality is maintained consistently across both training and test datasets. The error distribution in Figure 3 shows well-controlled deviations throughout the computational domain.

Table 3. Error and timing comparison for example 3 (Backward Kolmogorov Equation) using different methods

Method	Present Method		ADM [40]	HRBF [41]
	Train	Test		
Relative L_2	1.122994e-09	1.209307e-09	5.61e-08	5.9848e-04
RMS Error	3.161359e-09	3.316910e-09	-	6.2988e-04
MAE Error	2.280776e-09	2.682391e-09	-	5.1315e-04
Solver Time (s)	0.076		-	-
Test Eval Time (s)	0.516		-	-



(a) LSSVR-Legendre solution (b) Exact solution (c) Error distribution

Figure 3. Comparison of solutions for Example 3 (Backward Kolmogorov equation). Subfigure (a) shows the approximate solution obtained with the proposed LSSVR-Legendre method, (b) presents the exact analytical solution, and (c) displays the pointwise error across the domain.

4 Conclusion

This research successfully demonstrates the effectiveness of combining least squares support vector regression with Legendre kernels for solving linear FokkerPlanck equations. We developed a direct solution approach using LSSVR specifically tailored for these equations. The method solves the PDE without requiring spatialtemporal discretization and achieves high accuracy due to the strong approximation properties of Legendre kernels. The single-step LSSVR formulation provides computational advantages over iterative approaches and avoids the convergence issues of loss-based neural networks. Our numerical experiments confirm the reliability, generalization, and stability of the proposed framework across multiple test problems.

While the proposed framework offers several advantages, it also has some limitations. The current formulation has been tested primarily on linear one-dimensional problems, and its direct extension to nonlinear or high-dimensional FokkerPlanck equations may face challenges related to kernel expressiveness and computational scalability. The computational cost, though lower than training deep neural networks, still scales as $O(N^3)$ with the number of collocation points, which may restrict its use in very large-scale simulations. In addition, the choice of kernel parameters and truncation degree in Legendre expansions can significantly influence performance, and a systematic strategy for selecting these parameters is still an open problem.

Future research will therefore focus on extending this approach to nonlinear and higher-dimensional FokkerPlanck equations, where the benefits of global kernel representations could be even more significant. Exploring alternative kernel functions or hybrid formulations may further enhance flexibility and accuracy. To address computational scalability, parallel and GPU-based implementations will be investigated. Finally, applying the framework to real-world problems in finance, physics, and engineering will be an important step toward demonstrating its practical value beyond benchmark cases.

Authors' Contributions

The contributions of each author to this study are as follows: Maryam Babaei contributed to the research design, algorithm implementation, numerical experiments, and manuscript preparation. Kourosh Parand conceptualized the research framework, supervised the methodology development, provided theoretical guidance, and reviewed the manuscript.

Data Availability

All data generated or analyzed during this study are included in this published article. The numerical results and computational parameters are provided in the tables and can be reproduced using the described methodology and hyperparameters.

Conflicts of Interest

The authors declare that there is no conflict of interest.

Ethical Considerations

The authors have diligently addressed ethical concerns, such as informed consent, plagiarism, data fabrication, misconduct, falsification, double publication, redundancy, submission, and other related matters.

Funding

This research did not receive any grant from funding agencies in the public, commercial, or nonprofit sectors.

Acknowledgments

The authors acknowledge the computational resources provided by Shahid Beheshti University and thank the reviewers for their valuable comments that improved the quality of this manuscript.

References

[1] H. Risken, Fokker-planck equation. Springer, (1989).

[2] C. Gardiner, Stochastic methods, vol. 4. Springer Berlin Heidelberg, (2009).

[3] R. Cont and P. Tankov, Financial modelling with jump processes. Chapman and Hall/CRC, (2003).

[4] J. D. Murray, Mathematical biology: I. An introduction, vol. 17. Springer Science & Business Media, (2007).

[5] W. Coffey and Y. P. Kalmykov, The Langevin equation: with applications to stochastic problems in physics, chemistry and electrical engineering, vol. 27. World Scientific, (2012).

[6] C. Soize, Uncertainty quantification, vol. 23. Springer, (2017).

[7] T. D. Frank, Nonlinear Fokker-Planck equations: fundamentals and applications. Springer, (2005).

[8] G. A. Pavliotis, Stochastic processes and applications, 60, (2014).

[9] J. Chang and G. Cooper, A practical difference scheme for fokker-planck equations, Journal of Computational Physics, 6(1), 1–16, (1970).

[10] P. Kumar and S. Narayanan, Solution of fokker-planck equation by finite element and finite difference methods for nonlinear systems, Sadhana, 31(4), 445–461, (2006).

[11] D. Gottlieb and S. A. Orszag, Numerical analysis of spectral methods: theory and applications. SIAM, (1977).

[12] E. Platen and N. Bruti-Liberati, Numerical solution of stochastic differential equations with jumps in finance, vol. 64. Springer Science & Business Media, (2010).

[13] B. Oksendal, Stochastic differential equations: an introduction with applications. Springer Science & Business Media, (2013).

[14] J. A. Suykens and J. Vandewalle, Least squares support vector machine classifiers, vol. 9. Springer, (1999).

[15] S. Mehrkanoon, S. Mehrkanoon, and J. A. Suykens, Parameter estimation of delay differential equations: an integration-free ls-svm approach, Communications in Nonlinear Science and Numerical Simulation, 19(4), 830–841, (2014).

[16] A. Shidfar, A. Molabahrami, A. Babaei, and A. Yazdanian, A series solution of the nonlinear volterra and fredholm integro-differential equations, Communications in Nonlinear Science and Numerical Simulation, 15(2), 205–215, (2010).

[17] Z. Behdani and M. Darehmiraki, Enhancing kernel ridge regression models with compact support wendland functions, Analytical and Numerical Solutions for Nonlinear Equations, 9(1), 34–47, (2025).

[18] J. A. Rad, K. Parand, and S. Chakraverty, Learning with fractional orthogonal kernel classifiers in support vector machines: Theory, algorithms and applications. Springer, (2023).

[19] R. Pourgholi, H. Dana, and S. H. Tabasi, Solving an inverse heat conduction problem using genetic algorithm: sequential and multi-core parallelization approach, Applied Mathematical Modelling, 38(7-8), 1948–1958, (2014).

[20] H. Dana Mazraeh, M. Kalantari, S. H. Tabasi, A. Afzal Aghaei, Z. Kalantari, and F. Fahimi, Solving fredholm integral equations of the second kind using an improved cuckoo optimization algorithm, Analytical and Numerical Solutions for Nonlinear Equations, 7(1), 33–52, (2022).

[21] H. Dana Mazraeh, K. Parand, H. Farahani, and S. Kheradpisheh, An improved imperialist competitive algorithm for solving an inverse form of the huxley equation, Iranian Journal of Numerical Analysis and Optimization, 14(3), 681–707, (2024).

[22] E. Mirzabeigi, S. Rezaee, and K. Parand, Lyam: Robust non-convex optimization for stable learning in noisy environments, arXiv preprint arXiv:2507.11262, (2025).

[23] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, 378, 686–707, (2019).

[24] K. M. Mohammadi, M. Babaei, Z. Hajimohammadi, and K. Parand, A new numerical method for solving neuro-cognitive models via chebyshev deep neural network (cdnn), Applications of Mathematics, 1–19, (2025).

[25] S. Wang, Y. Teng, and P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks, SIAM Journal on Scientific Computing, 43(5), A3055–A3081, (2021).

[26] E. Mirzabeigi, R. Salehi, and K. Parand, Bridgenet: A hybrid, physics-informed machine learning framework for solving high-dimensional fokker-planck equations, arXiv preprint arXiv:2506.04354, (2025).

[27] E. Mirzabeigi, M. Babaei, A. H. Karami, S. Rezaee, R. Salehi, and K. Parand, Physics-informed lane-emden solvers using lynx-net: Implementing radial basis functions in kolmogorov representation, Astronomy and Computing, 100997, (2025).

[28] J. P. Boyd, Chebyshev and Fourier spectral methods. Courier Corporation, (2001).

[29] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, Spectral methods: fundamentals in single domains. Springer, (2006).

[30] J. Shen, T. Tang, and L.-L. Wang, Spectral methods: algorithms, analysis and applications, vol. 41. Springer Science & Business Media, (2011).

[31] S. Mehrkanoon and J. A. Suykens, Learning solutions to partial differential equations using ls-svm, 159, 105–116, (2015).

[32] M. Razzaghi and S. Yousefi, The legendre wavelets operational matrix of integration, International journal of systems science, 32(4), 495–502, (2001).

[33] M. Babaei, A. A. Aghaei, Z. Kazemi, M. Jamshidi, R. Ghaderi, and K. Parand, Solving a class of thomas–fermi equations: A new solution concept based on physics-informed machine learning, 225, 716–730, (2024).

[34] S. Mirzaei and A. Shokri, A pseudo-spectral approach to solving the fractional cable equation using lagrange polynomials, Analytical and Numerical Solutions for Nonlinear Equations, 9(1), 20–33, (2025).

[35] R. Burden, J. Faires, and A. Burden, Numerical Analysis. Cengage Learning, (2015).

[36] P. Ahadian and K. Parand, Support vector regression for the temperature-stimulated drug release, 165, 112871, (2022).

[37] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2623–2631, (2019).

[38] M. Lakestani and M. Dehghan, Numerical solution of fokker-planck equation using the cubic b-spline scaling functions, Numerical Methods for Partial Differential Equations: An International Journal, 25(2), 418–429, (2009).

[39] H. Dana Mazraeh, P. Motaharinezhad, N. Daneshian, and K. Parand, Solving the fokker-planck equation with neural networks: A performance improvement approach, Analytical and Numerical Solutions for Nonlinear Equations, 9(1), 1–11, (2025).

[40] M. Tatari, M. Dehghan, and M. Razzaghi, Application of the adomian decomposition method for the fokker–planck equation, Mathematical and Computer Modelling, 45(5-6), 639–650, (2007).

[41] S. Kazem, J. Rad, and K. Parand, Radial basis functions methods for solving fokker–planck equation, Engineering Analysis with Boundary Elements, 36(2), 181–189, (2012).