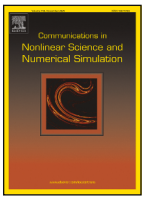
Commun Nonlinear Sci Numer Simulat 152 (2026) 109414

Contents lists available at ScienceDirect

Communications in Nonlinear Science and Numerical Simulation

journal homepage: www.elsevier.com/locate/cnsns



Research paper

J-PIKAN: A physics-informed KAN network based on Jacobi orthogonal polynomials for solving fluid dynamics

Xiong Xiong [a,b], Kang Lu [b,c], Zhuo Zhang [d], Zheng Zeng [b,c], Sheng Zhou [b,c], Zichen Deng [a,b,c,e], Rongchun Hu [b,c,*]

[a] School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an, 710072, PR China
[b] MIIT Key Laboratory of Dynamics and Control of Complex Systems, Northwestern Polytechnical University, Xi'an, 710072, PR China
[c] Department of Engineering Mechanics, Northwestern Polytechnical University, Xi'an, 710072, PR China
[d] College of Computer Science and Technology, National University of Defense Technology, Changsha, Hunan, 410073, PR China
[e] Department of Aeronautical Engineering, Northwestern Polytechnical University, Xi'an, 710072, PR China

ARTICLE INFO

Keywords:
Jacobi orthogonal polynomials
Kolmogorov–Arnold network
Physics-informed neural networks
Fluid dynamics
Partial differential equations

ABSTRACT

Traditional Physics-Informed Neural Networks (PINNs) based on multilayer perceptrons face optimization difficulties, spectral bias, and limited parameter efficiency when solving complex fluid dynamics problems. This work addresses these limitations by developing J-PIKAN, a physics-informed Kolmogorov-Arnold Network based on Jacobi orthogonal polynomials.We present a systematic comparison of different basis functions (Jacobi polynomials with various α,β parameters, Chebyshev, Legendre, Hermite, Fourier, B-spline, and Taylor) across five representative fluid dynamics benchmarks. Our comprehensive analysis reveals that Jacobi polynomials consistently achieve superior performance, delivering 1–2 orders of magnitude improvement in solution accuracy compared to baseline MLPs across different equation types. Through Hessian eigenvalue analysis, we demonstrate that J-PIKAN exhibits more favorable optimization characteristics with reduced numerical ill-conditioning during training. For high Reynolds number lid-driven cavity flows, J-PIKAN maintains superior accuracy while requiring only 50 % of the parameters compared to basic MLPs. J-PIKAN offers a promising framework for developing efficient and reliable deep learning-based solvers for fluid dynamics, demonstrating significant improvements in both accuracy and parameter efficiency while addressing numerical stability challenges associated with polynomial-based networks. Code will be made available at https://github.com/xgxgnpu/J-PIKAN upon acceptance of the paper.

1. Introduction

Partial differential equations (PDEs) are fundamental tools for modeling fluid dynamics systems [1], and their efficient and accurate solution is crucial for predicting the physical behavior of dynamic systems and obtaining high-fidelity physical information. Traditional numerical methods, such as finite differences, finite volumes, and finite elements, can provide precise solutions but often face the challenge of high computational costs when dealing with complex geometries, high-dimensional problems, and inverse problems.

* Corresponding author.
E-mail address: rongchun_hu@nwpu.edu.cn (R. Hu).

https://doi.org/10.1016/j.cnsns.2025.109414
Received 24 May 2025; Received in revised form 19 September 2025; Accepted 7 October 2025
Available online 11 October 2025

In recent years, Physics-Informed Neural Networks (PINNs) have shown significant potential in solving both forward and inverse PDE problems by embedding physical laws directly into the deep learning framework [2]. PINNs have been widely applied in various fields, including fluid dynamics [3–15], solid mechanics [16–18], and nonlinear dynamical systems [19–21,26]. Recent advances have explored various PINN variants, including fractional-order PINNs [22] for handling fractional differential equations, Bayesian PINNs [23] for uncertainty quantification with noisy data, and physics-informed neural operators [24,25] for learning solution operators across parameter spaces. Comprehensive reviews of these physics-informed machine learning approaches can be found in Karniadakis et al. [27], Cuomo et al. [28]. As a mesh-free method, PINNs utilize automatic differentiation [29] to construct a physics-informed loss function, offering a simple way to jointly solve both forward and inverse problems within the same optimization framework. However, MLP-based PINNs still face several inherent limitations including optimization difficulties [30,31], spectral bias, and limited parameter efficiency in complex nonlinear problems. These challenges often manifest as ill-conditioning issues during training [32], requiring specialized techniques such as pseudo-time stepping approaches [33] to achieve stable convergence.

To overcome these limitations, Kolmogorov–Arnold Networks (KANs), based on the Kolmogorov–Arnold representation theorem [34], have emerged. Unlike traditional MLPs, KANs replace fixed activation functions with learnable univariate functions, fundamentally altering the construction of neural networks. The development of learnable activation functions has been an active area of research, with parametric activation functions showing promise for improved neural network performance [35]. Adaptive activation functions have also been demonstrated to accelerate convergence in physics-informed neural networks [36]. Various basis functions have been explored for KAN implementations, including B-splines [37], radial basis functions [38], wavelet basis functions [39], Chebyshev basis functions [40–42], and Fourier series. Recent studies have also investigated specialized KAN architectures, such as Kurkova Kolmogorov–Arnold Networks (KKANs) [43] for improved approximation capabilities, and explored Chebyshev-based KANs for turbulence modeling applications [44].

Most relevant to our work, orthogonal polynomial-based KANs have been preliminarily explored for PDE problems. The theoretical foundation of orthogonal polynomials [45] and their applications in computational methods, particularly in spectral approaches [46], have been well established in the literature. Notably, Shukla et al. [47] recently provided a comprehensive comparison between MLP and KAN representations for differential equations, including the application of Jacobi polynomial-based KANs to problems such as lid-driven cavity flow. This pioneering work established the feasibility of using various orthogonal polynomials in KAN architectures and demonstrated promising initial results in comparing KAN versus MLP performance. Additionally, Chebyshev polynomial-based physics-informed KANs have been successfully applied to fluid mechanics problems [41], and turbulent thermal convection inference [44], while KAN-based approaches have been applied to sparse data model discovery in ordinary differential equations [48] and forward PDE problems in solid mechanics [49]. Moreover, hidden fluid mechanics problems [50] have demonstrated the potential of neural networks in learning velocity and pressure fields from flow visualizations, further motivating the development of advanced neural architectures for fluid dynamics applications. Recent work has also shown promise in handling discontinuous problems using physics-informed approaches [51], expanding the scope of neural network-based PDE solvers.

While these initial explorations have demonstrated the potential of polynomial-based KANs, several important gaps remain in our understanding. First, systematic comparison of different basis functions within a unified experimental framework has been limited. Second, comprehensive analysis of the general Jacobi polynomial family with different parameter combinations (α, β) across diverse problem types is lacking. Third, in-depth investigation of optimization characteristics and performance scaling across challenging fluid dynamics scenarios remains unexplored. Recent advances in physics-informed neural networks have achieved remarkable accuracy on standard benchmarks through improved optimization strategies [52,53], utilizing various optimizers such as Adam [54] and L-BFGS [55], and high-performance architectures like PirateNets [56] have shown strong performance on complex flows, but the systematic evaluation of polynomial-based KANs in this context requires further investigation.

Building upon this foundation, the present study provides the first comprehensive and systematic investigation of Jacobi polynomial-based physics-informed KANs across a diverse range of fluid dynamics problems. Our contributions include: (1) systematic comparison of different basis functions (Jacobi polynomials with various α,β parameters, Chebyshev [40], Legendre, Hermite, Fourier, B-spline, and Taylor) within a unified experimental framework; (2) in-depth analysis of the general Jacobi polynomial family [57] with comprehensive parameter studies providing practical selection guidance; (3) establishment of performance benchmarks across five representative fluid dynamics problems spanning different mathematical characteristics from 1D nonlinear equations to high Reynolds number 2D flows, including standard benchmark cases such as the lid-driven cavity flow [58]; (4) empirical investigation of optimization dynamics through Hessian eigenvalue analysis; and (5) demonstration of superior parameter efficiency achieving comparable or better accuracy with significantly fewer parameters than conventional approaches.

By leveraging Jacobi orthogonal polynomials as learnable activation functions, our approach fundamentally addresses key limitations of traditional neural network architectures in solving fluid dynamics problems. The orthogonality of Jacobi polynomials [59–61] significantly reduces interference between basis functions, effectively alleviating numerical stiffness during the optimization process, thereby stabilizing training and accelerating convergence. Through the combined expression of orthogonal basis functions and basis coefficients, J-PIKAN not only maintains the network's expressive capability but also drastically reduces the number of parameters, enhancing computational efficiency. Most importantly, the completeness and tunable parameters of the Jacobi polynomial system provide the network with powerful function approximation capabilities, making it particularly suitable for solving complex fluid dynamics problems with strong nonlinearities.

The organization of this paper is as follows: Section 2 introduces Jacobi orthogonal polynomials and presents the principles of the Jacobian Orthogonal Polynomial-based Physics-Informed Kolmogorov–Arnold Network (J-PIKAN), analyzing optimization characteristics theoretically; Section 3 presents numerical results, including five benchmark problems and comprehensive result analysis; finally, Section 4 concludes the paper.

2. Jacobian orthogonal polynomial-based physics-informed Kolmogorov–Arnold network

2.1. Problem description

We investigate nonlinear partial differential equations arising in fluid dynamics, where the general form of the governing equations subject to initial and boundary conditions is given by:

$$F_\theta[u](x) = f(x), \quad x \in \Omega, \tag{2.1}$$

$$B_\theta[u](x) = b(x), \quad x \in \partial\Omega, \tag{2.2}$$

$$I_\theta[u](x) = g(x), \quad x \in \Omega_0, \tag{2.3}$$

where x represents the spatiotemporal coordinates, u is the unknown solution, and θ denotes the neural network parameters. Here, f is the source term, b is the boundary term, and g is the initial condition. F and B are general nonlinear differential and boundary operators, respectively, and I is the initial condition operator. The region Ω is a bounded domain, $\partial\Omega$ represents its boundary, and Ω_0 is the spatial domain at the initial time. In this work, we primarily focus on approximating the solution u to the differential equation using a neural network u_θ, where θ represents the parameters. The initial condition loss, boundary condition loss, and physics-informed loss functions are constructed through automatic differentiation [29], and the optimal θ is determined by minimizing these loss functions through optimization.

2.2. Jacobian orthogonal polynomials

Prior to presenting the formulation of the Jacobian Orthogonal Polynomial-based Physics-Informed Kolmogorov–Arnold Network, we establish the fundamental mathematical preliminaries. Central to our framework is the theory of orthogonal polynomials, which constitutes a complete system of polynomials $p_n(x)$ defined on a compact interval $[a, b]$ satisfying the orthogonality relation:

$$\int_a^b w(x)p_m(x)p_n(x)dx = \delta_{mn}c_n, \tag{2.4}$$

$$c_n \equiv \int_a^b w(x)[p_n(x)]^2 dx, \tag{2.5}$$

where $w(x)$ is the weight function, and δ_{mn} is the Kronecker delta function. When $c_n = 1$, these polynomials are not only orthogonal but also normalized orthogonal polynomials.

Orthogonal polynomials have highly useful properties in solving differential equations [46,57] and physical problems. Just as Fourier series provide a convenient method for expanding periodic functions in terms of linearly independent terms, orthogonal polynomials offer a natural approach for solving, expanding, and interpreting many important differential equations. Orthogonal polynomials can be readily generated using the Gram-Schmidt orthogonalization process.

In our framework, we employ the classical Jacobi polynomials—a fundamental class of orthogonal polynomials that form a complete basis in weighted L^2 spaces. These polynomials, denoted as $J_n^{(\alpha,\beta)}(x)$, are defined on the finite interval $[-1, 1]$ with parameters $\alpha, \beta > -1$, and constitute a particularly versatile family of orthogonal polynomials. The Jacobi polynomials satisfy the following three-term recurrence relation:

$$J_n^{(\alpha,\beta)}(x) = -\frac{(\alpha+n-1)(\beta+n-1)(\alpha+\beta+2n)}{n(\alpha+\beta+n)(\alpha+\beta+2n-2)} J_{n-2}^{(\alpha,\beta)}(x) \tag{2.6}$$

$$+ \frac{(\alpha+\beta+2n-1)\left\{\alpha^2-\beta^2+x(\alpha+\beta+2n)(\alpha+\beta+2n-2)\right\}}{2n(\alpha+\beta+n)(\alpha+\beta+2n-2)} \tag{2.7}$$

$$\times J_{n-1}^{(\alpha,\beta)}(x), \quad n = 2, 3, \ldots, \tag{2.8}$$

This recurrence relation demonstrates that the Jacobi polynomial $J_n^{(\alpha,\beta)}(x)$ is constructed from the previous two terms, $J_{n-1}^{(\alpha,\beta)}(x)$ and $J_{n-2}^{(\alpha,\beta)}(x)$. Specifically,

$$J_0^{(\alpha,\beta)}(x) = 1, \quad J_1^{(\alpha,\beta)}(x) = \frac{\alpha+\beta+2}{2}x + \frac{\alpha-\beta}{2}. \tag{2.9}$$

Their orthogonality is defined by the L_2 inner product:

$$\int_{-1}^{1}(1-x)^\alpha(1+x)^\beta J_m^{(\alpha,\beta)}(x)J_n^{(\alpha,\beta)}(x)dx = \frac{2^{\alpha+\beta+1}}{2n+\alpha+\beta+1}\frac{\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}{\Gamma(n+\alpha+\beta+1)n!}\delta_{nm}. \tag{2.10}$$

Jacobi polynomials exhibit some special cases. When $\alpha = \beta = 0$, the Jacobi polynomials reduce to the Legendre polynomials:

$$P_0(x) = 1, \quad P_1(x) = x, \tag{2.11}$$

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \quad n \geq 1. \tag{2.12}$$

When $\alpha = \beta = -\frac{1}{2}$, the Jacobi polynomials reduce to the Chebyshev polynomials:

$$T_0(x) = 1, \quad T_1(x) = x, \tag{2.13}$$

Table 1
Mathematical expressions of Jacobi polynomials $J_n^{(\alpha,\beta)}(x)$ for polynomial degrees $n = 0, 1, 2, 3, 4$ with parameter combinations $\alpha = \beta = 2, 1, 0, -0.5$, serving as basis functions in J-PIKAN.

n	$\alpha = \beta = 2$ (Jacobi)	$\alpha = \beta = 1$ (Jacobi)	$\alpha = \beta = 0$ (Legendre)	$\alpha = \beta = -0.5$ (Chebyshev)
0	1	1	1	1
1	$\frac{5}{2}x$	$\frac{3}{2}x$	x	x
2	$\frac{1}{8}(21x^2 - 7)$	$\frac{1}{8}(15x^2 - 5)$	$\frac{1}{2}(3x^2 - 1)$	$2x^2 - 1$
3	$\frac{1}{16}(105x^3 - 105x)$	$\frac{1}{8}(35x^3 - 30x)$	$\frac{1}{2}(5x^3 - 3x)$	$4x^3 - 4x$
4	$\frac{1}{128}(945x^4 - 1260x^2 + 315)$	$\frac{1}{128}(315x^4 - 420x^2 + 105)$	$\frac{1}{8}(35x^4 - 30x^2 + 3)$	$8x^4 - 8x^2 + 1$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geq 1. \tag{2.14}$$

Table 1 and Fig. 1 present the expressions and geometric characteristics of the Jacobi polynomials under different parameter conditions. By comparing the Jacobi polynomials for various values of $\alpha = \beta$ (2,1,0 and -0.5), two important special cases are considered: the Legendre polynomials ($\alpha = \beta = 0$) and the first-kind Chebyshev polynomials ($\alpha = \beta = -0.5$). As observed, with the increase in the degree n, the expressions of the polynomials exhibit regular complex variations and display different oscillatory behaviors over the interval $[-1, 1]$.

The Jacobi orthogonal polynomial family offers unique advantages in neural network construction: with its complete orthogonality and tunable parameter characteristics (α and β), it can be expanded as a basis function, providing a richer nonlinear feature representation compared to traditional activation functions such as Tanh and ReLU. Additionally, its recurrence relations enable efficient computation, and the orthogonality property helps alleviate ill-conditioning during training. These advantages make neural networks built on this polynomial family particularly well-suited for addressing complex fluid dynamics problems and enhancing model interpretability.

2.3. Jacobian physics-informed Kolmogorov–Arnold network

In this section, we introduce a function representation method based on the Kolmogorov–Arnold representation theorem and construct our Jacobian Physics-Informed Kolmogorov–Arnold Network (J-PIKAN) based on this theorem. Fig. 2 illustrates the comprehensive computational framework and optimization process of J-PIKAN.

According to the Kolmogorov–Arnold representation theorem, any continuous function $u_\theta : \mathbb{R}^n \to \mathbb{R}$ can be represented as:

$$u_\theta(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{2n+1} \Phi_i \left(\sum_{j=1}^{n} \phi_{ij}(x_j) \right), \tag{2.15}$$

where $x_1, x_2, \ldots, x_n$ are the input spatiotemporal coordinates, $\Phi_i : \mathbb{R} \to \mathbb{R}$ are continuous univariate functions in the outer layer, and $\phi_{ij} : \mathbb{R} \to \mathbb{R}$ are continuous univariate functions in the inner layer. Here, $1 \leq i \leq 2n + 1$ and $1 \leq j \leq n$ represent the indices of the functions.

Unlike traditional MLPs that use fixed activation functions, KANs use learnable univariate functions. In the original KAN model, $\phi(x) = w_b b(x) + w_s \text{spline}(x)$, where $b(x)$ is a commonly used activation function such as Tanh, Sigmoid, or ReLU, and spline(x) is a B-spline function, which is also referred to as a learnable activation function. The parameters w_b and w_s are weights used to balance the contributions of the fixed activation function and the B-spline function.

For the task of solving partial differential equations, the B-spline-based KAN has been shown to be inefficient and to have relatively low accuracy. To more accurately and efficiently solve complex nonlinear fluid dynamics equations in the fluid domain, we propose a new KAN model, the J-PIKAN.

The J-PIKAN leverages the excellent orthogonality and approximation capabilities of Jacobi orthogonal polynomials by introducing them into the learnable univariate functions. For a deep KAN architecture with L layers, each univariate function $\phi_{ij}^{(l)}(x)$ in layer l is defined as:

$$\phi_{ij}^{(l)}(x) = \sum_{n=0}^{N} c_{n,ij}^{(l)} J_n^{(\alpha,\beta)}(\tanh(x)), \tag{2.16}$$

where $J_n^{(\alpha,\beta)}(x)$ represents the Jacobi polynomial of degree n with parameters α and β, $c_{n,ij}^{(l)}$ are the trainable coefficients corresponding to each polynomial basis and each edge connecting neuron j in layer l to neuron i in layer $l + 1$, and N is the maximum polynomial degree. The $\tanh(x)$ function serves as a normalization function to ensure that inputs to the Jacobi polynomials are within the required orthogonality interval $[-1, 1]$ and provides numerical stability. The selection of Jacobi polynomial parameters α and β significantly impacts network performance. Based on extensive experimental validation across diverse fluid dynamics problems, we find that $\alpha = \beta = 1$ and $\alpha = \beta = 2$ consistently achieve optimal performance. While α and β need not be equal in general, and the optimal parameter combination constitutes a complex optimization problem in itself, this work focuses on symmetric cases ($\alpha = \beta$) for systematic comparison. Future research could explore asymmetric parameter combinations and adaptive parameter selection strategies to further enhance J-PIKAN's performance across different problem domains.

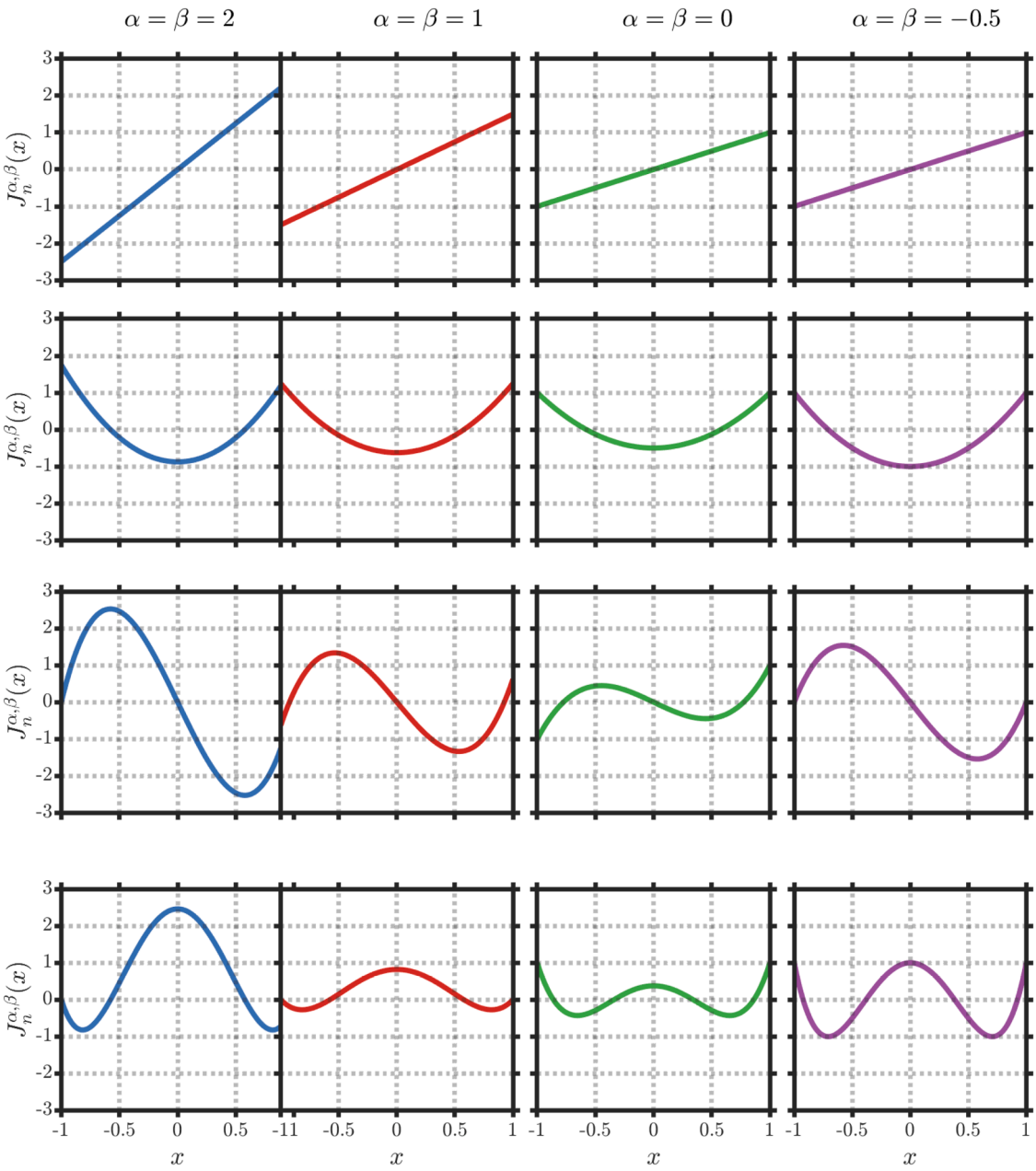


Fig. 1. Comprehensive visualization of Jacobi polynomial basis functions $J_n^{(\alpha,\beta)}(x)$ demonstrating the effect of polynomial order ($n = 1, 2, 3, 4$) and parameter combinations ($\alpha = \beta = 2, 1, 0, -0.5$) over the normalized interval $x \in [-1, 1]$. The plots illustrate the orthogonal characteristics and varying oscillatory behaviors that provide the mathematical foundation for J-PIKAN's learnable activation functions, with special cases including Legendre polynomials ($\alpha = \beta = 0$) and Chebyshev polynomials ($\alpha = \beta = -0.5$).

The Jacobi polynomials satisfy the recurrence relation:

$$J_n^{(\alpha,\beta)}(x) = (A_n x + B_n) J_{n-1}^{(\alpha,\beta)}(x) + C_n J_{n-2}^{(\alpha,\beta)}(x), \tag{2.17}$$

with the following expressions for A_n, B_n, and C_n:

$$A_n = \frac{(2n + \alpha + \beta - 1)(2n + \alpha + \beta)}{2n(n + \alpha + \beta)} \tag{2.18}$$

$$B_n = \frac{(2n + \alpha + \beta - 1)(\alpha^2 - \beta^2)}{2n(n + \alpha + \beta)(2n + \alpha + \beta - 2)} \tag{2.19}$$

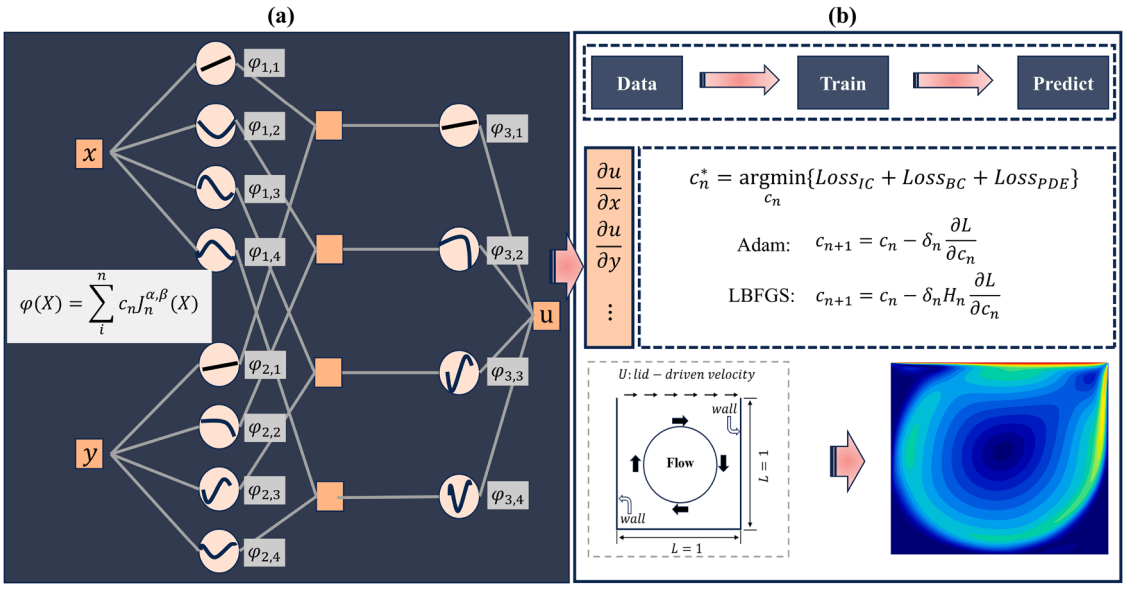


Fig. 2. Comprehensive architecture diagram of the Jacobi Physics-Informed Kolmogorov–Arnold Network (J-PIKAN) framework showcasing the fundamental departure from conventional multilayer perceptrons. Unlike traditional MLPs with fixed activation functions, J-PIKAN implements learnable activation functions through Jacobi polynomial expansions: $\phi_{ij}^{(l)}(X) = \sum_{n=0}^{N} c_{n,ij}^{(l)} J_n^{(\alpha,\beta)}(\tanh(X))$, where $c_{n,ij}^{(l)}$ represent trainable polynomial coefficients and $\tanh(X)$ ensures input normalization to the orthogonality interval $[-1, 1]$. Panel (a) illustrates a fundamental core layer template of J-PIKAN, where each ϕ-node embodies a learnable function. Deep J-PIKAN networks are constructed through multiple compositional nesting of this single core layer template, enabling hierarchical feature extraction across the entire network depth through the composition $u_\theta(\mathbf{x}) = \mathbf{\Phi}^{(L-1)} \circ \mathbf{\Phi}^{(L-2)} \circ \ldots \circ \mathbf{\Phi}^{(0)}(\mathbf{x})$. Panel (b) depicts the integrated physics-constrained optimization pipeline incorporating initial conditions, boundary conditions, and partial differential equation residual constraints.

$$C_n = \frac{-2(n+\alpha-1)(n+\beta-1)(2n+\alpha+\beta)}{2n(n+\alpha+\beta)(2n+\alpha+\beta-2)} \tag{2.20}$$

For the deep J-PIKAN network with L layers, the forward propagation is defined by the layer-wise transformations:

$$\mathbf{x}^{(l+1)} = \mathbf{\Phi}^{(l)}(\mathbf{x}^{(l)}), \quad l = 0, 1, \ldots, L-1 \tag{2.21}$$

where $\mathbf{x}^{(0)}$ is the input, $\mathbf{x}^{(L)}$ is the output, and $\mathbf{\Phi}^{(l)}$ represents the function transformation at layer l. Each component of the layer transformation is defined as:

$$x_i^{(l+1)} = \sum_{j=1}^{n_l} \phi_{ij}^{(l)}(x_j^{(l)}), \quad i = 1, 2, \ldots, n_{l+1} \tag{2.22}$$

where n_l and n_{l+1} are the dimensions of layers l and $l+1$, respectively.

The input and output relationship of the lth layer can be expressed in matrix form as:

$$\mathbf{x}^{(l+1)} = \underbrace{\begin{pmatrix} \phi_{1,1}^{(l)}(\cdot) & \phi_{1,2}^{(l)}(\cdot) & \cdots & \phi_{1,n_l}^{(l)}(\cdot) \\ \phi_{2,1}^{(l)}(\cdot) & \phi_{2,2}^{(l)}(\cdot) & \cdots & \phi_{2,n_l}^{(l)}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n_{l+1},1}^{(l)}(\cdot) & \phi_{n_{l+1},2}^{(l)}(\cdot) & \cdots & \phi_{n_{l+1},n_l}^{(l)}(\cdot) \end{pmatrix}}_{\mathbf{\Phi}^{(l)}} \mathbf{x}^{(l)}, \tag{2.23}$$

where $\mathbf{\Phi}^{(l)}$ is the function matrix of the lth layer.

The complete deep J-PIKAN network can then be expressed as the composition:

$$u_\theta(\mathbf{x}) = \mathbf{\Phi}^{(L-1)} \circ \mathbf{\Phi}^{(L-2)} \circ \ldots \circ \mathbf{\Phi}^{(1)} \circ \mathbf{\Phi}^{(0)}(\mathbf{x}) \tag{2.24}$$

Since the Jacobi polynomials require the input to be within the interval $[-1, 1]$ to maintain their orthogonality properties, the input normalization using the hyperbolic tangent function is essential:

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}. \tag{2.25}$$

This normalization function serves dual purposes: ensuring the Jacobi polynomials operate within their defined orthogonality interval and providing numerical stability during computation. Alternative scaling strategies, such as min-max normalization or variance-mean normalization, may also be used.

By substituting the solution u_θ into the J-PIKAN network and leveraging automatic differentiation, we obtain the total loss function for J-PIKAN:

$$\mathcal{L}_{J-PIKAN}(\theta) = \lambda_{IC}\mathcal{L}_{IC}(\theta) + \lambda_{BC}\mathcal{L}_{BC}(\theta) + \lambda_{PDE}\mathcal{L}_{PDE}(\theta), \tag{2.26}$$

where λ_{IC}, λ_{BC}, and λ_{PDE} are the weighting factors for the initial condition, boundary condition, and PDE residual losses, respectively. These weighting factors are crucial for balancing the relative importance of different physical constraints during optimization. Each component is defined as follows:

$$\mathcal{L}_{PDE}(\theta) = \frac{1}{N_f}\sum_{j=1}^{N_f} \|(F_\theta[u_\theta](x_j^f) - f_j)\|^2, \tag{2.27}$$

$$\mathcal{L}_{BC}(\theta) = \frac{1}{N_b}\sum_{l=1}^{N_b} \|(B_\theta[u_\theta](x_l^b) - b_l)\|^2, \tag{2.28}$$

$$\mathcal{L}_{IC}(\theta) = \frac{1}{N_i}\sum_{k=1}^{N_i} \|(I_\theta[u_\theta](x_k^i) - g_k)\|^2. \tag{2.29}$$

The selection of appropriate weighting factors λ_{IC}, λ_{BC}, and λ_{PDE} is problem-dependent and typically requires careful tuning. Common strategies include: (1) setting equal weights ($\lambda_{IC} = \lambda_{BC} = \lambda_{PDE} = 1$) as a baseline; (2) using adaptive weighting schemes that adjust weights during training based on the relative magnitudes of different loss components; (3) employing problem-specific weights based on physical understanding, where boundary conditions might be weighted more heavily for boundary-dominated problems.To ensure fair comparison across all test cases in this study, we employ equal weights ($\lambda_{IC} = \lambda_{BC} = \lambda_{PDE} = 1$) for all benchmark problems unless otherwise specified.

To optimize $\mathcal{L}_{J-PIKAN}(\theta)$, we follow the approach outlined in the original PINNs literature by first applying the Adam optimizer [54] for global stochastic optimization and then using the L-BFGS optimizer [55] for local optimization. This two-stage optimization strategy leverages Adam's global search capability and robustness in handling non-convex optimization problems, while L-BFGS accelerates convergence near the optimal solution using approximate Hessian matrix information, allowing for more precise solutions. In the first stage, the Adam optimizer effectively explores the parameter space through adaptive learning rates and momentum mechanisms, overcoming the influence of local minima and providing a good initial solution. In the second stage, the L-BFGS optimizer, as a quasi-Newton method, refines the solution with high accuracy in regions close to the optimal solution. This combination of optimization techniques has been shown to significantly improve both the training efficiency and prediction accuracy of physics-informed neural networks in practice.

Jacobi polynomials, as basis functions, offer three prominent advantages that make them uniquely advantageous for constructing the Physics-Informed KAN network. First, in terms of expressiveness, since they form a complete orthogonal basis for the $L^2[-1,1]$ space, any square-integrable function can be uniquely represented by a Jacobi polynomial series, with coefficients that are optimally computed. This guarantees more accurate function approximation. Second, in terms of numerical stability, the orthogonality between basis functions keeps the condition number of the Gram matrix on the order of $\mathcal{O}(1)$, avoiding the numerical instability of B-splines in higher-order cases. Third, regarding parameter efficiency, Jacobi polynomials exhibit spectral convergence for analytic functions, with each coefficient corresponding directly to specific frequency information, thus avoiding the redundancy of control points in B-splines and the low parameter efficiency of MLPs. This ensures higher computational precision and faster convergence at the same parameter scale.

It is important to note that we construct the Jacobi orthogonal polynomials as a deep KAN network, rather than using a single-layer combination of basis functions and coefficients, as in traditional numerical methods. Based on the theoretical framework outlined earlier, we have implemented a deep neural network architecture based on Jacobi polynomials. Each layer of the network, referred to as the JacobiKANLayer, is constructed based on the Kolmogorov–Arnold representation theorem, with each univariate function $\phi_{ij}^{(l)}(x)$ expanded using Jacobi polynomials. Specifically, for each layer of the network, the input is first normalized to the $[-1,1]$ interval using the tanh function to maintain orthogonality properties, then the Jacobi polynomials are used as basis functions for expansion. The entire network consists of multiple such layers connected in series, forming a deep structure represented as $u_\theta(\mathbf{x}) = \mathbf{\Phi}^{(L-1)} \circ \mathbf{\Phi}^{(L-2)} \circ \ldots \circ \mathbf{\Phi}^{(0)}(\mathbf{x})$.

This design maintains the advantages of traditional deep learning while incorporating the mathematical properties of orthogonal polynomials. The number of parameters in each layer l is given by $n_l \times n_{l+1} \times (N+1)$, where n_l and n_{l+1} are the input and output dimensions of the layer, and N is the maximum polynomial degree, and the complexity of the model can be flexibly controlled by adjusting the polynomial degree N. The use of orthogonal polynomial basis functions and tanh normalization ensures numerical stability in the computations. Each coefficient $c_{n,ij}^{(l)}$ corresponds to the basis coefficient of polynomial degree n for the edge connecting neuron j in layer l to neuron i in layer $l+1$, carrying clear mathematical meaning. By combining different degrees of orthogonal polynomials, the network can effectively approximate complex nonlinear functions. This structure provides a theoretically grounded and practically efficient framework for solving complex physical problems.

Although the theoretical advantages of Jacobi polynomials (such as complete orthogonal bases, numerical stability, and spectral convergence) cannot be fully inherited in the deep neural network framework, combining these superior properties with the powerful expressive capability of deep learning still results in significant benefits. In terms of expressiveness, their strong mathematical properties and multi-scale feature capture ability enhance the network's function approximation capability. In terms of numerical computation, their lower correlation between basis functions improves training stability. In terms of parameter efficiency, the properties

of basis functions corresponding to different scale features allow the network to efficiently learn complex physical problems with fewer parameters, while the clear physical and mathematical meaning of the basis functions also enhances the interpretability of the network. The synergistic effects of these advantages enable the Jacobi-based J-PIKAN to achieve a good balance between accuracy, stability, and computational efficiency when handling complex nonlinear flow field computations.

2.4. Empirical analysis of optimization characteristics in J-PIKAN

In the training process of Physics-Informed Neural Networks, numerical ill-conditioning is a key factor affecting model convergence and accuracy. This section analyzes the numerical ill-conditioning characteristics of the J-PIKAN model when addressing complex fluid dynamics problems and explores the advantages of Jacobi orthogonal polynomials in alleviating these issues.

Based on our empirical observations across multiple fluid dynamics benchmarks, we conjecture that J-PIKAN exhibits more favorable optimization dynamics compared to traditional MLPs. The following analysis provides theoretical motivation for this empirical finding, though rigorous mathematical proof remains an open question for future investigation.

Similar to MLP-based PINNs, the training process of J-PIKAN can be described by the gradient flow system:

$$\frac{d\theta}{dt} = -\nabla_\theta \mathcal{L}_{PDE}(\theta) - \nabla_\theta \mathcal{L}_{BC}(\theta) - \nabla_\theta \mathcal{L}_{IC}(\theta) \tag{2.30}$$

This corresponds to the gradient descent process of the total loss function $\mathcal{L}_{J-PIKAN}(\theta)$. As is well-known, the largest eigenvalue of the Hessian matrix $\sigma_{\max}(\nabla^2_\theta \mathcal{L}(\theta))$ determines the fastest timescale in the gradient flow system and directly limits the learning rate that can be used in gradient descent. According to numerical analysis theory, to ensure the stability of the forward Euler discretization (i.e., gradient descent update), the learning rate must satisfy the condition $\eta < 2/\sigma_{\max}(\nabla^2_\theta \mathcal{L}(\theta))$. When the condition number of the Hessian matrix is large, this constraint becomes particularly strict, resulting in the need for extremely small learning rates during the training process, which significantly reduces the convergence speed. Existing research indicates that the ill-conditioning in PINNs primarily arises from the Hessian matrix of the PDE loss function, where the eigenvalue distribution is highly uneven, with a few extremely large eigenvalues and many eigenvalues close to zero, leading to an abnormally high condition number.

To better understand the numerical ill-conditioning issue in the J-PIKAN model, we examine the changes in the PDE loss function during gradient descent to reveal the relationship between stiffness and training difficulty. Consider the nth training step, where the parameter update follows:

$$\theta_{n+1} = \theta_n - \eta \nabla_\theta \mathcal{L}_{PDE}(\theta_n) \tag{2.31}$$

where η is the learning rate. By performing a second-order Taylor expansion of the PDE loss function $\mathcal{L}_{PDE}(\theta)$ around θ_n and combining it with the parameter update formula, we obtain:

$$\mathcal{L}_{PDE}(\theta_{n+1}) - \mathcal{L}_{PDE}(\theta_n) = -\eta \|\nabla_\theta \mathcal{L}_{PDE}(\theta_n)\|_2^2 + \frac{1}{2}\eta^2 \nabla_\theta \mathcal{L}_{PDE}(\theta_n)^T \nabla^2_\theta \mathcal{L}_{PDE}(\xi) \nabla_\theta \mathcal{L}_{PDE}(\theta_n) \tag{2.32}$$

where ξ is a point between θ_n and θ_{n+1}. Through eigenvalue decomposition, the above expression can be rewritten as Wang et al. [30]:

$$\mathcal{L}_{PDE}(\theta_{n+1}) - \mathcal{L}_{PDE}(\theta_n) = \eta \|\nabla_\theta \mathcal{L}_{PDE}(\theta_n)\|_2^2 \left(-1 + \frac{1}{2}\eta \sum_{i=1}^{M} \lambda_i^{PDE} y_i^2 \right) \tag{2.33}$$

where λ_i^{PDE} denotes the ith eigenvalue of the Hessian matrix $\nabla^2_\theta \mathcal{L}_{PDE}(\xi)$, and y_i represents the projection of the normalized gradient in the direction of the corresponding eigenvector.

When the gradient flow is ill-conditioned, the eigenvalue distribution of the Hessian matrix $\nabla^2_\theta \mathcal{L}_{PDE}(\xi)$ becomes highly uneven, with some eigenvalues being significantly large. In cases where the largest eigenvalue is extremely large, we observe that even when η is set to a small value, the condition $\mathcal{L}_{PDE}(\theta_{n+1}) - \mathcal{L}_{PDE}(\theta_n) > 0$ may still occur, meaning that the gradient descent method cannot effectively reduce the PDE loss function value. This phenomenon is particularly prevalent in the training of PINNs. In the J-PIKAN model, we conjecture that due to the orthogonality properties of the Jacobi polynomials, the largest eigenvalue of the Hessian matrix of the PDE loss function may be lower than that of MLPs, potentially alleviating the ill-conditioning problem during J-PIKAN training and enabling faster model convergence. This analysis suggests that the orthogonality properties of Jacobi polynomials may contribute to improved conditioning, though the complex interactions in deep networks require further theoretical investigation.

3. Numerical experiments

In this section, we systematically evaluate J-PIKAN's efficacy through a comprehensive benchmark suite spanning diverse fluid dynamics phenomena. Our validation encompasses five canonical problems: the one-dimensional Burgers equation for nonlinear shock wave dynamics, the Korteweg-de Vries (KdV) equation with third-order derivatives and soliton solutions for multi-scale assessment, the Taylor–Green vortex for two-dimensional unsteady flow evolution, the Kovasznay flow for steady-state validation, and lid-driven cavity flows at elevated Reynolds numbers ($Re = 1000, 2500, 4000$) for complex boundary layer and vortical structure resolution. All simulations are implemented in PyTorch framework and executed on NVIDIA RTX 4090 GPUs, employing cosine annealing learning rate decay schedules to enhance convergence stability throughout the optimization process.

Our evaluation focuses on three critical performance aspects: (a) computational accuracy in capturing complex flow physics and discontinuous phenomena, (b) parameter efficiency achieving superior performance with reduced network complexity, and (c)

Table 2
Training configuration summary for J-PIKAN with Jacobi polynomials ($\alpha = \beta = 2$) across seven fluid dynamics benchmarks: network architectures with parameter counts 4300–6600, evaluated over 5 random initialization seeds.

Test Cases	N. Params	Adam Iter.	L-BFGS Iter.	Rel. l_2 Error
Burgers	4300	1×10^4	10,000	1.1×10^{-4}
KdV	6300	1×10^4	10,000	3.7×10^{-3}
Taylor–Green vortex	6600	1×10^4	10,000	u: 4.6×10^{-3} v: 6.5×10^{-3} p: 8.1×10^{-3}
Kovasznay flow	6500	–	10,000	u: 1.1×10^{-3} v: 6.3×10^{-3} p: 7.6×10^{-3}
Lid-driven cavity ($Re = 1000$)	6500	–	9000	1.3×10^{-2}
Lid-driven cavity ($Re = 2500$)	6500	–	18,000	4.6×10^{-2}
Lid-driven cavity ($Re = 4000$)	6500	–	18,000	5.2×10^{-2}

optimization conditioning through Hessian eigenvalue analysis during training convergence. The relative L_2 error metric is defined as Relative L_2 Error $= \frac{\|u_{\text{pred}} - u_{\text{exact}}\|_2}{\|u_{\text{exact}}\|_2}$, where u_{pred} represents the neural network prediction and u_{exact} denotes the analytical or high-fidelity numerical reference solution. This metric provides quantitative assessment of solution accuracy across all benchmark problems.

Table 2 summarizes the comprehensive test suite configuration and performance overview. J-PIKAN with Jacobi polynomials ($\alpha = \beta = 2$) demonstrates exceptional accuracy across all benchmarks, achieving relative L_2 errors of 1.1×10^{-4} for the Burgers equation and 1.1×10^{-3} for Kovasznay flow. Remarkably, even for the challenging high Reynolds number lid-driven cavity flow ($Re = 2500$), our method maintains acceptable accuracy at 4.6×10^{-2} error level. These results establish J-PIKAN's versatility in handling both one-dimensional shock phenomena and complex multi-dimensional nonlinear flow dynamics with superior computational efficiency.

3.1. Burgers equation

We begin by using the Burgers equation to test J-PIKAN's ability to capture nonlinear shock waves. The Burgers' equation is a fundamental model in fluid dynamics describing one-dimensional fluid motion. It is widely applied in traffic flow simulation, acoustics, gas dynamics, and shock wave propagation. The equation combines a nonlinear convection term (uu_x) with a diffusion term (u_{xx}), enabling the capture of nonlinear wave phenomena. Solving the Burgers' equation presents computational challenges, as the nonlinear term makes analytical solutions difficult, especially at small viscosity coefficients ($\frac{0.01}{\pi}$), where the solution develops sharp shock waves in finite time. This challenges the stability and accuracy of numerical methods. The equation is:

$$u_t + uu_x - \frac{0.01}{\pi} u_{xx} = 0, \quad x \in [-1, 1], \quad t \in [0, 1], \tag{3.1}$$

with initial conditions $u(0, x) = -\sin(\pi x)$ and periodic boundary conditions $u(t, -1) = u(t, 1)$.

For the J-PIKAN loss function, we randomly sample 10,000 points within the spatiotemporal domain $[0, 1] \times [-1, 1]$ for the PDE residual loss, 100 points at $t = 0$ for the initial condition loss, and 100 points at the boundaries $x = \pm 1$ for the boundary condition loss. The weighted combination of these losses is optimized using the Adam optimizer for 2×10^4 iterations with initial learning rate 10^{-3} and cosine annealing, followed by L-BFGS optimization for up to 20,000 iterations.

To evaluate J-PIKAN's performance on nonlinear shock problems, we compare it with various Physics-Informed Neural Networks using different basis functions, including Jacobi polynomials with parameters ($\alpha = \beta = 2, 1, 0, -0.5$), Hermite polynomials, Fourier, B-spline, and standard MLPs. As shown in Table 3, with network architecture [2, 20, 20, 1], J-PIKAN with Jacobi polynomials ($\alpha = \beta = 2$) achieved the highest accuracy (relative L_2 error of 5.8×10^{-4}) with only 2300 parameters. This significantly outperforms the baseline MLP (5081 parameters, error $= 5.3 \times 10^{-3}$) and other methods, demonstrating superior parameter efficiency. Fig. 3 shows that J-PIKAN not only exhibits excellent overall accuracy but also accurately captures steep gradients in the shock wave region, demonstrating excellent numerical stability and feature resolution capabilities.

The architecture sensitivity study in Table 4 shows that model accuracy improves significantly when network width increases from 15 to 20 neurons. However, further increasing width to 25 yields minimal improvement, providing guidance for practical architecture selection.

Fig. 4 compares different physics-informed models for the Burgers equation. The heatmaps show spatiotemporal solution distributions with corresponding L_2 errors. J-PIKAN (particularly $\alpha = \beta = 2$) accurately captures solution dynamics, especially in the shock region near $t = 0.5$. Compared to the analytical solution, J-PIKAN predicts shock location and intensity accurately while preserving solution smoothness. This performance stems from Jacobi polynomials' advantages in handling nonlinear problems and their multi-scale expressiveness.

The training dynamics in Fig. 5 show that J-PIKAN converges faster and maintains stable training with higher accuracy than baseline methods. This advantage is explained by the Hessian eigenvalue evolution in Fig. 6. During training, the maximum eigenvalue for Jacobi basis functions remains smaller than MLPs. At convergence, Jacobi functions reach maximum eigenvalues below 1×10^4, while MLP eigenvalues approach 1×10^5. This creates a smoother loss landscape for J-PIKAN with better-conditioned optimization

Table 3
Training protocol comparison for Burgers equation solvers: unified optimization scheme with Adam optimizer (2×10^4 iterations, learning rate 10^{-3}, cosine annealing) + L-BFGS (maximum 20,000 iterations), sampling strategy of 10,000 PDE residual points, 100 initial condition points, 100 boundary points, comparing network architectures [2,20,20,1] for polynomial-based models vs. [2,40,40,40,40,1] for MLP, evaluated over 5 random seeds.

Model	Network Architecture	N. Params	Rel. l_2 Error
Jacobi($\alpha = \beta = 2$)	[2, 20, 20, 1]	2300	$(5.8 \pm 1.7) \times 10^{-4}$
Jacobi($\alpha = \beta = 1$)	[2, 20, 20, 1]	2300	$(1.8 \pm 0.5) \times 10^{-3}$
Chebyshev	[2, 20, 20, 1]	2300	$(5.6 \pm 1.4) \times 10^{-3}$
Legendre	[2, 20, 20, 1]	2300	$(5.5 \pm 1.8) \times 10^{-3}$
Hermite	[2, 20, 20, 1]	2300	$(8.6 \pm 2.1) \times 10^{-3}$
Fourier	[2, 20, 20, 1]	3721	$(1.6 \pm 0.6) \times 10^{-1}$
B-spline	[2, 20, 20, 1]	4140	$(2.1 \pm 0.7) \times 10^{-1}$
MLP	[2,40,40,40,40,1]	5081	$(5.3 \pm 1.9) \times 10^{-3}$

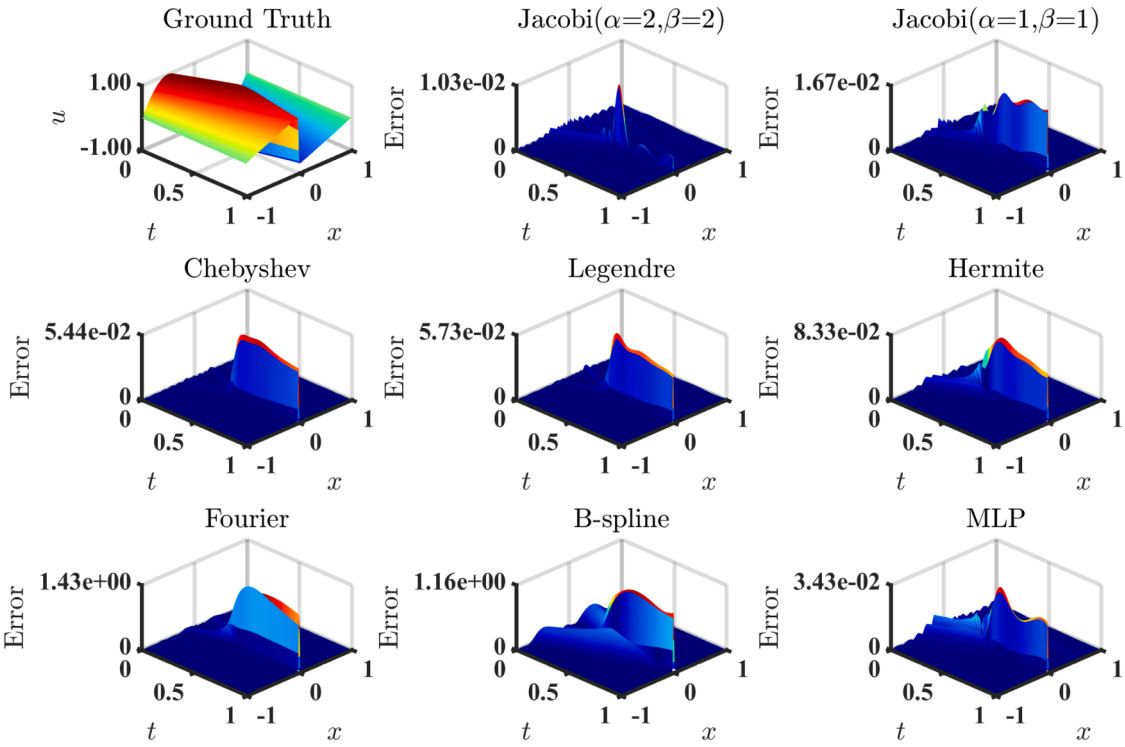


Fig. 3. Three-dimensional spatiotemporal error distribution analysis for the Burgers equation solution, providing quantitative comparison of prediction accuracy between J-PIKAN and alternative physics-informed neural network approaches. The heatmaps reveal error magnitudes across the computational domain $x \in [-1, 1]$, $t \in [0, 1]$, with particular emphasis on the challenging shock wave propagation region near $t = 0.5$.

Table 4
Architecture sensitivity study for J-PIKAN on Burgers equation: network width scaling (10, 15, 20, 25 neurons) with fixed depth (4 layers) and polynomial degree (3), using standardized training protocol (Adam + L-BFGS optimization), sampling configuration of 10,000 domain points plus 200 boundary/initial condition points, parameter scaling analysis from 1320 to 7800 trainable weights, evaluated over 5 random initialization seeds.

Width	Depth	Degree	N. Params	Rel. l_2 Error
10	4	3	1320	$(7.8 \pm 2.3) \times 10^{-3}$
15	4	3	2880	$(4.1 \pm 1.2) \times 10^{-4}$
20	4	3	5040	$(3.5 \pm 0.9) \times 10^{-4}$
25	4	3	7800	$(6.7 \pm 1.7) \times 10^{-4}$

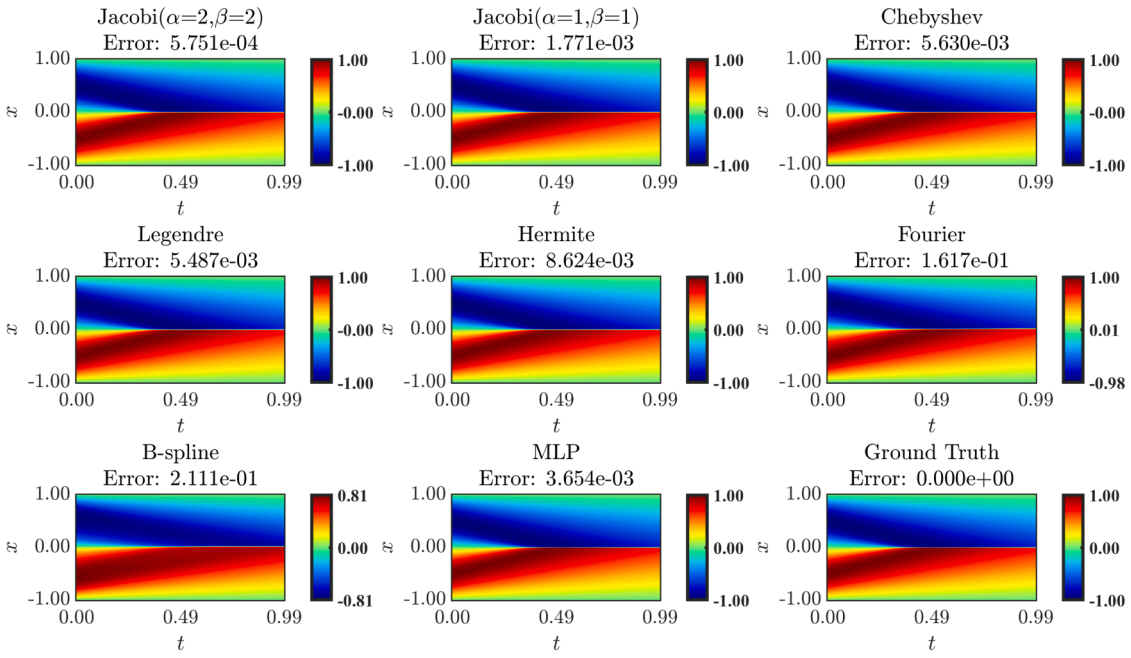


Fig. 4. Comprehensive performance comparison of physics-informed neural networks for Burgers equation prediction showcasing ground truth solution versus predictions from multiple basis function implementations: Jacobi polynomials with varying parameters ($\alpha = \beta = 2, 1, 0, -0.5$), Hermite polynomials, Fourier series, B-spline functions, and traditional multilayer perceptrons. Each subplot displays the spatiotemporal solution field with corresponding relative l_2 error metrics, clearly demonstrating J-PIKAN's superior accuracy in capturing shock wave dynamics, particularly in the challenging high-gradient regions where discontinuous behavior emerges.

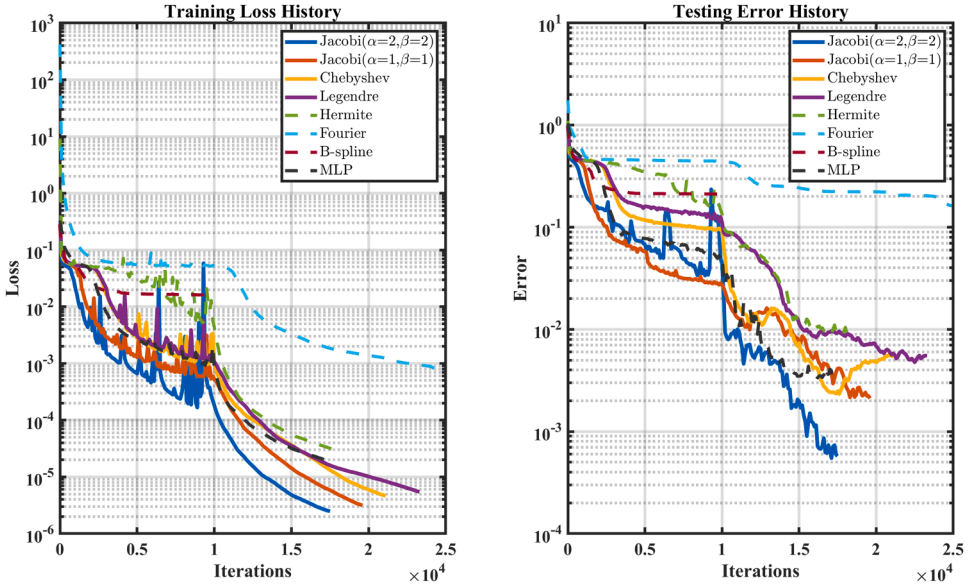


Fig. 5. Training loss and testing error histories for the Burgers equation: Convergence comparison among different basis functions including Jacobi polynomials ($\alpha = \beta = 2, 1, 0, -0.5$), some other classical orthogonal polynomials, and MLPs.

and improved stability. Error analysis in Fig. 3 confirms that J-PIKAN maintains low, uniform errors across the computational domain, including high-gradient shock regions.

The results demonstrate that J-PIKAN exhibits significant advantages over conventional methods when solving the nonlinear Burgers' equation in terms of parameter efficiency, prediction accuracy, and optimization conditioning. These findings validate the method's effectiveness and provide insights for developing efficient scientific computing approaches. The advantages are particularly pronounced for complex nonlinear phenomena such as shock waves, offering practical value for engineering flow problems.

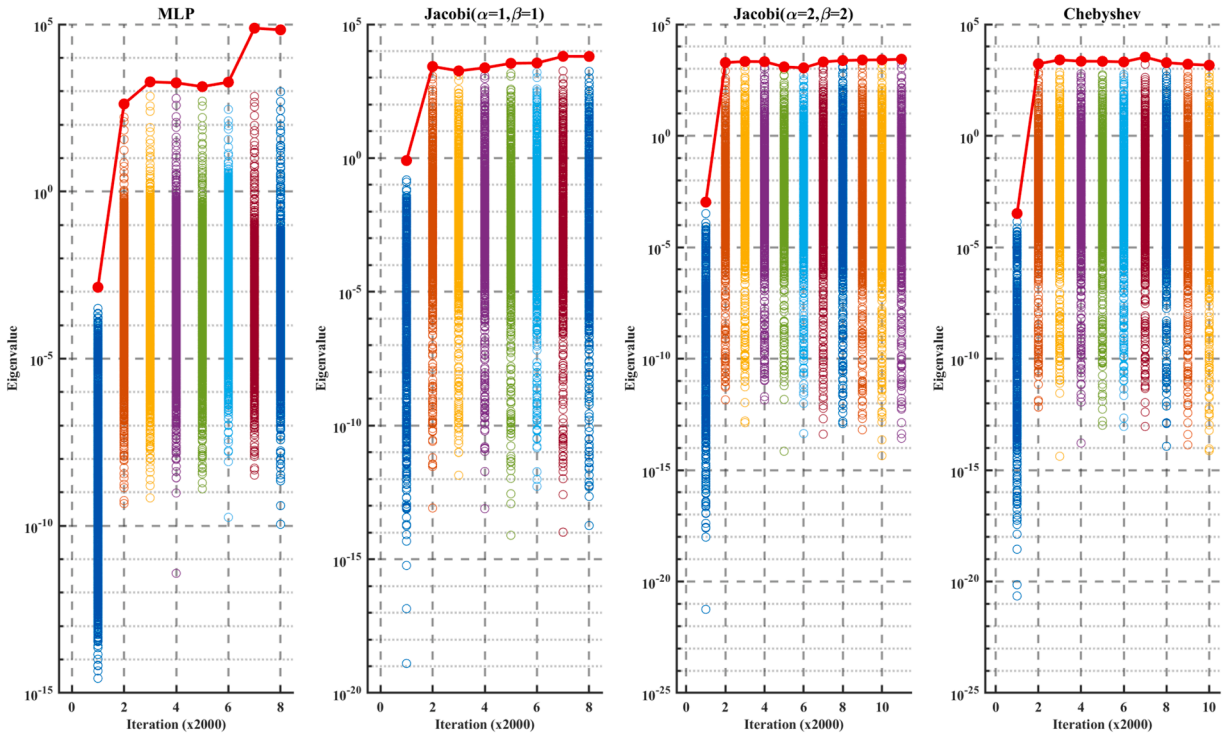


Fig. 6. Evolution of Hessian eigenvalues during training for the Burgers equation: Comparison of eigenvalue distributions across different models (MLP, Jacobi, and Chebyshev), with red dots indicating maximum eigenvalues.

3.2. KdV equation

To evaluate the performance of J-PIKAN in handling the nonlinear soliton solution of the KdV equation with higher-order partial derivatives, we conducted a comparison with several physics-informed neural network models built using different basis functions. The KdV (Korteweg-de Vries) equation and its boundary conditions are as follows:

$$u_t + 6uu_x + u_{xxx} = 0, \quad x \in [-1, 1], \quad t \in [0, 1] \tag{3.2}$$

$$u(0, x) = \cos(\pi x) \tag{3.3}$$

$$u(t, -1) = u(t, 1) \tag{3.4}$$

$$u_x(t, -1) = u_x(t, 1) \tag{3.5}$$

$$u_{xx}(t, -1) = u_{xx}(t, 1) \tag{3.6}$$

Here, $u(t, x)$ represents the amplitude of the wave, where the three terms in the first equation represent time evolution (u_t), nonlinear effects ($6uu_x$), and dispersion effects (u_{xxx}). The initial condition is a cosine wave, and the boundary conditions are periodic, requiring the function values and their first and second derivatives to be continuous at the boundaries. This equation describes the propagation of one-dimensional shallow water waves and can generate stable soliton solutions.

To assess J-PIKAN's ability to capture soliton solutions, we compared models with identical architecture [2, 20, 20, 20, 20, 1] using different basis functions: Jacobi polynomials ($\alpha = \beta = 2, 1$), Chebyshev, Legendre, and traditional MLPs. Table 5 shows that Jacobi-based models achieved superior accuracy, with Legendre polynomials performing best (relative L_2 error: 5.4×10^{-3}). Despite using nearly double the parameters (10,401 vs. 5040), the MLP error (6.9×10^{-2}) remained significantly higher, demonstrating J-PIKAN's parameter efficiency.

Table 6 examines width sensitivity for Jacobi polynomials ($\alpha = \beta = 2$) with fixed depth (4 layers) and polynomial degree (4). Increasing width from 15 to 25 neurons substantially reduces error from 6.5×10^{-2} to 3.4×10^{-3}, highlighting the importance of network expressiveness while considering computational overhead.

Fig. 7 demonstrates that J-PIKAN ($\alpha = \beta = 2$) achieves 5.1×10^{-3} relative L_2 error with architecture [2, 20, 20, 1], outperforming MLPs (6.9×10^{-2}) by an order of magnitude. Crucially, J-PIKAN maintains soliton amplitude during temporal evolution $t \in [0, 1]$, while MLPs exhibit significant propagation errors.

Training dynamics in Fig. 8 show J-PIKAN reaches 10^{-4} training loss within 5×10^3 iterations, while MLPs remain at 10^{-2}. After 2×10^4 iterations, Jacobi models ($\alpha = \beta = 1$) achieve 4.5×10^{-3} test error—a 90.8 % reduction compared to MLPs (7.2×10^{-2}).

Table 5
KdV equation training configuration: network architectures [2,20,20,20,20,1] for polynomial models and [2,50,50,50,50,50,1] for MLP comparison, sampling strategy with 10,000 spatiotemporal domain points, periodic boundary conditions including first and second derivative constraints, two-stage optimization (Adam 1×10^4 iterations + L-BFGS refinement), results from 5 random initialization seeds.

Model	Network Architecture	N. Params	Rel. l_2 Error
Jacobi($\alpha = \beta = 2$)	[2, 20, 20, 20, 20, 1]	5040	$(8.6 \pm 2.4) \times 10^{-3}$
Jacobi($\alpha = \beta = 1$)	[2, 20, 20, 20, 20, 1]	5040	$(1.0 \pm 0.3) \times 10^{-2}$
Chebyshev	[2, 20, 20, 20, 20, 1]	5040	$(6.7 \pm 1.8) \times 10^{-3}$
Legendre	[2, 20, 20, 20, 20, 1]	5040	$(5.4 \pm 1.3) \times 10^{-3}$
MLP	[2, 50, 50, 50, 50,50, 1]	10,401	$(6.9 \pm 2.8) \times 10^{-2}$

Table 6
Network architecture optimization for KdV equation using J-PIKAN: systematic width study (15, 20, 25 neurons) with fixed depth (4 layers) and polynomial degree (4), parameter count scaling from 3600 to 9750, evaluated over 5 random seeds.

Width	Depth	Degree	N. Params	Rel. l_2 Error
15	4	4	3600	$(6.5 \pm 2.1) \times 10^{-2}$
20	4	4	6300	$(5.1 \pm 1.4) \times 10^{-3}$
25	4	4	9750	$(3.4 \pm 1.0) \times 10^{-3}$

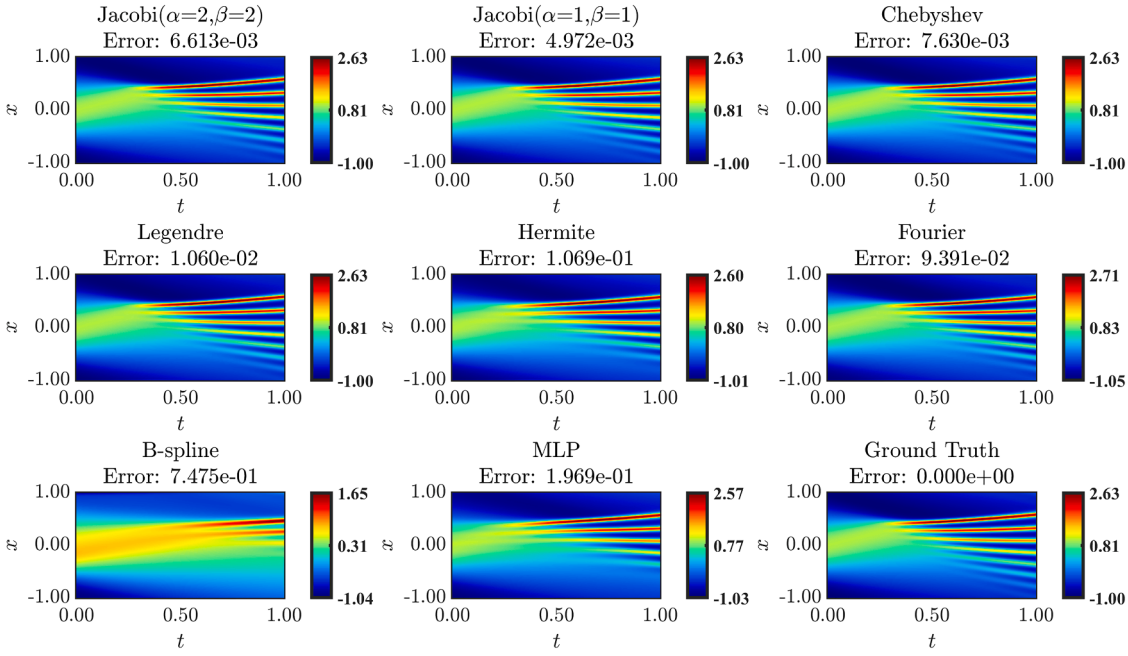


Fig. 7. Comparison of different physics-informed models for KdV equation prediction: Ground truth vs. Jacobi polynomials ($\alpha = \beta = 2, 1, 0, -0.5$),Hermite polynomials, Fourier series, B-spline, and MLP, with their respective l_2 errors.

Fig. 9 reveals that J-PIKAN maintains uniform low errors across the computational domain, including high-gradient soliton regions. Fig. 10 shows that Jacobi basis functions develop smaller maximum eigenvalues than MLPs during training, creating smoother loss landscapes with better conditioning and optimization stability.

The results demonstrate J-PIKAN's significant advantages for solving the nonlinear KdV equation in parameter efficiency, accuracy, and optimization conditioning. These findings validate the method's effectiveness for complex nonlinear phenomena and highlight its practical value for fluid dynamics applications.

3.3. Taylor–Green vortex

To further demonstrate the effectiveness of J-PIKAN in solving high-dimensional flow field problems, we chose the Taylor–Green vortex as a test case. The Taylor–Green vortex is a classic fluid dynamics problem frequently used in computational fluid dynamics

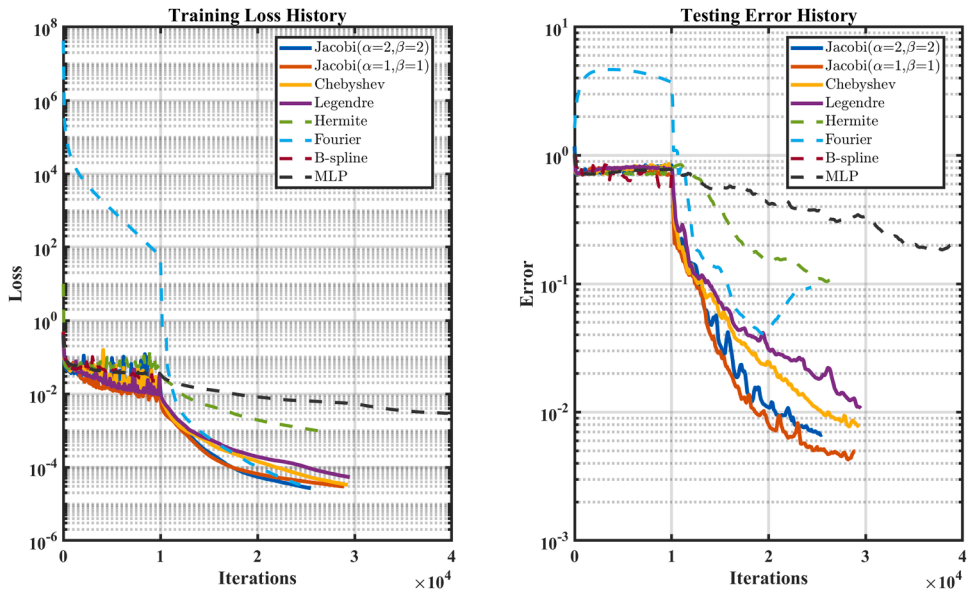


Fig. 8. Training loss and testing error histories for the KdV equation: Convergence comparison among different basis functions including Jacobi polynomials ($\alpha = \beta = 2$, $\alpha = \beta = 1$), some classical orthogonal polynomials, and MLPs.

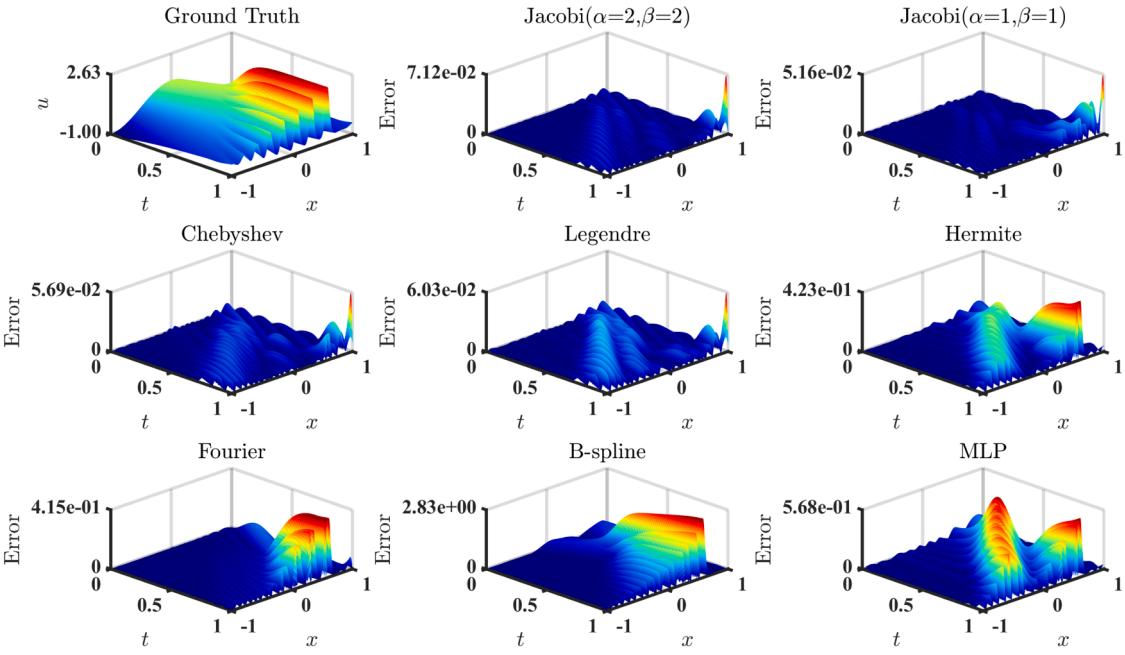


Fig. 9. Spatiotemporal distribution of solution errors for the KdV equation: Comparison of different methods against ground truth.

to evaluate the performance of numerical methods. It can be described using the two-dimensional unsteady incompressible Navier–Stokes equations:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{3.7}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{3.8}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{3.9}$$

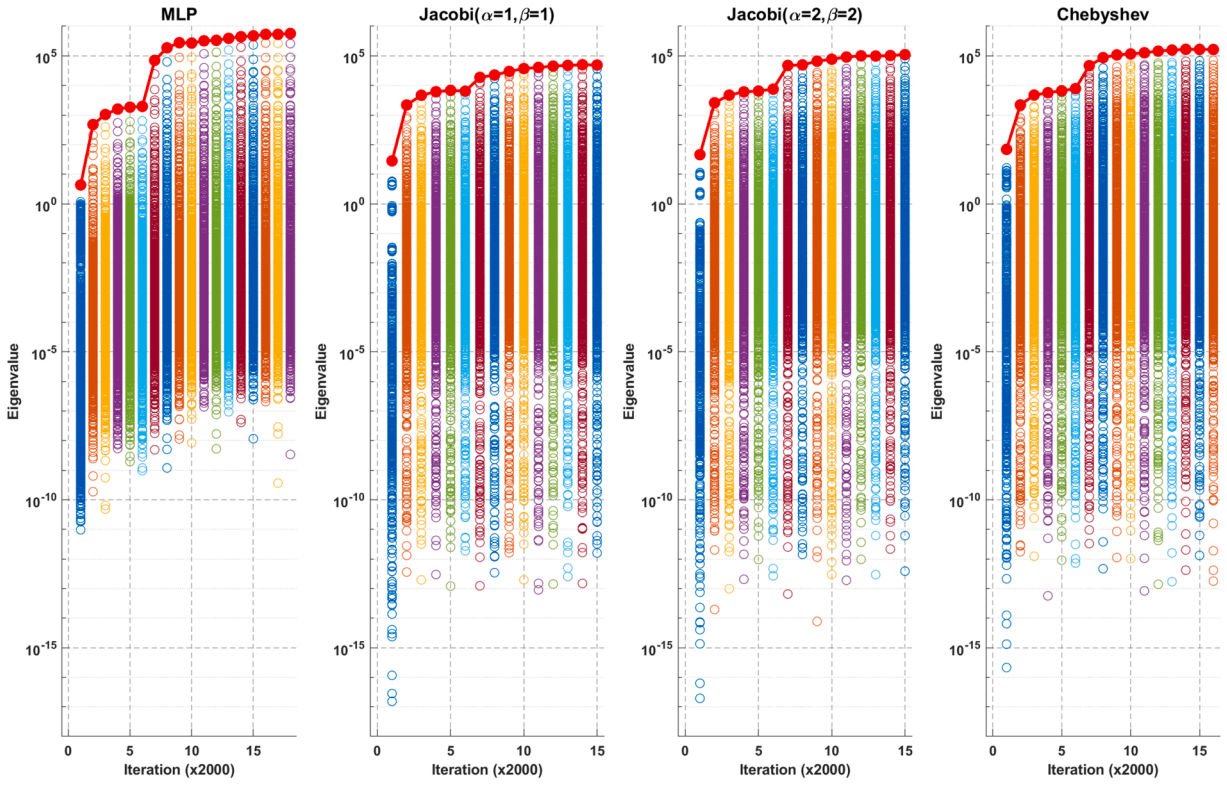


Fig. 10. Evolution of Hessian eigenvalues during training for the KdV equation: Comparison of eigenvalue distributions across different models (MLP, Jacobi, and Chebyshev), with red dots indicating maximum eigenvalues.

Here, u and v are the velocity components in the x and y directions, respectively, p is the pressure, ρ is the fluid density, and ν is the kinematic viscosity. Eqs. (3.7) and (3.8) represent the momentum equations in the x and y directions, and Eq. (3.9) is the continuity equation.

For the Taylor–Green vortex problem, the Reynolds number $\mathrm{Re} = \frac{UL}{\nu}$ is typically used to characterize the flow, where U and L are the characteristic velocity and length scale, respectively. The analytical solution for this problem in two dimensions is given by:

$$\begin{aligned} u(x,y,t) &= -\cos(\pi x)\sin(\pi y)\exp\left(-\frac{2\pi^2 t}{\mathrm{Re}}\right), \\ v(x,y,t) &= \sin(\pi x)\cos(\pi y)\exp\left(-\frac{2\pi^2 t}{\mathrm{Re}}\right), \\ p(x,y,t) &= -\frac{1}{4}[\cos(2\pi x)+\cos(2\pi y)]\exp\left(-\frac{4\pi^2 t}{\mathrm{Re}}\right), \end{aligned} \tag{3.10}$$

We set the Reynolds number $\mathrm{Re} = 100$, with $x, y \in [-\pi, \pi] \times [-\pi, \pi]$ and the time range $t \in [0, 1]$. To construct the initial loss, boundary loss, PDE loss, and total loss function, we sampled 256×100 points on each boundary in the x and y directions, 100×100 points at the initial time $t = 0$, and $20,000$ random points in the interior region. During the optimization of the total loss, we first used the Adam optimizer for 2000 iterations and then applied the L-BFGS optimizer for an additional 2000 iterations.

Table 7 compares different basis-function-based physics-informed KAN models for the Taylor–Green vortex using unified architecture [3, 30, 30, 30, 30, 3]. We evaluated solution accuracy via relative L_2 errors for velocity components (u, v) and pressure field (p). J-PIKAN demonstrated significant advantages: compared to Taylor polynomials with identical parameters, accuracy improved by nearly two orders of magnitude. Even against Fourier basis functions using more parameters (23,163 vs. 14,400), prediction error reduced by approximately 40×, demonstrating superior parameter efficiency for unsteady high-dimensional flow problems.

Figs. 11–13 show velocity and pressure predictions for the Taylor–Green vortex. Chebyshev polynomials (a special case of Jacobi polynomials) exhibit outstanding performance across all physical quantities. The comprehensive performance superiority includes: u-velocity error of 4.1×10^{-3} (significantly better than Hermite's 2.4×10^{-1}); v-velocity error of 5.3×10^{-3} (nearly two orders of magnitude improvement over B-spline and Taylor at 2.8×10^{-1}); pressure prediction error of 1.2×10^{-2} (40× and 80× lower than Fourier and Taylor, respectively). These results highlight the crucial role of basis function selection and J-PIKAN's capability for high-dimensional flow problems.

Table 7
Taylor–Green vortex training framework (Re = 100): unified architecture [3,30,30,30,30,3] across eight basis function implementations, comprehensive sampling with 256 × 100 boundary points per spatial direction, 100 × 100 initial condition grid, 20,000 interior random points, two-stage optimization (Adam 2000 iterations + L-BFGS 2000 iterations), multi-physics validation for velocity components (u, v) and pressure field (p), evaluated over 5 random initialization seeds.

Model	Network Architecture	N. Params	Rel. l_2 Error u	Rel. l_2 Error v	Rel. l_2 Error p
Jacobi($\alpha = \beta = 2$)	[3, 30, 30, 30, 30, 3]	14,400	$(4.1 \pm 1.1) \times 10^{-3}$	$(5.3 \pm 1.5) \times 10^{-3}$	$(1.2 \pm 0.4) \times 10^{-2}$
Jacobi($\alpha = \beta = 1$)	[3, 30, 30, 30, 30, 3]	14,400	$(5.5 \pm 1.6) \times 10^{-3}$	$(6.4 \pm 1.9) \times 10^{-3}$	$(1.8 \pm 0.6) \times 10^{-2}$
Chebyshev	[3, 30, 30, 30, 30, 3]	14,400	$(5.3 \pm 1.3) \times 10^{-3}$	$(5.9 \pm 1.7) \times 10^{-3}$	$(1.6 \pm 0.5) \times 10^{-2}$
Legendre	[3, 30, 30, 30, 30, 3]	14,400	$(6.5 \pm 1.8) \times 10^{-3}$	$(6.4 \pm 2.0) \times 10^{-3}$	$(4.3 \pm 1.4) \times 10^{-2}$
Hermite	[3, 30, 30, 30, 30, 3]	14,400	$(2.4 \pm 0.8) \times 10^{-1}$	$(2.5 \pm 0.9) \times 10^{-1}$	$(6.9 \pm 2.3) \times 10^{-1}$
Fourier	[3, 30, 30, 30, 30, 3]	23,163	$(1.6 \pm 0.6) \times 10^{-1}$	$(1.1 \pm 0.4) \times 10^{-1}$	$(4.9 \pm 1.7) \times 10^{-1}$
B-spline	[3, 30, 30, 30, 30, 3]	14,400	$(4.7 \pm 1.6) \times 10^{-1}$	$(2.8 \pm 0.9) \times 10^{-1}$	$(6.2 \pm 2.1) \times 10^{-1}$
Taylor	[3, 30, 30, 30, 30, 3]	14,400	$(2.4 \pm 0.8) \times 10^{-1}$	$(2.8 \pm 1.0) \times 10^{-1}$	$(9.6 \pm 3.2) \times 10^{-1}$

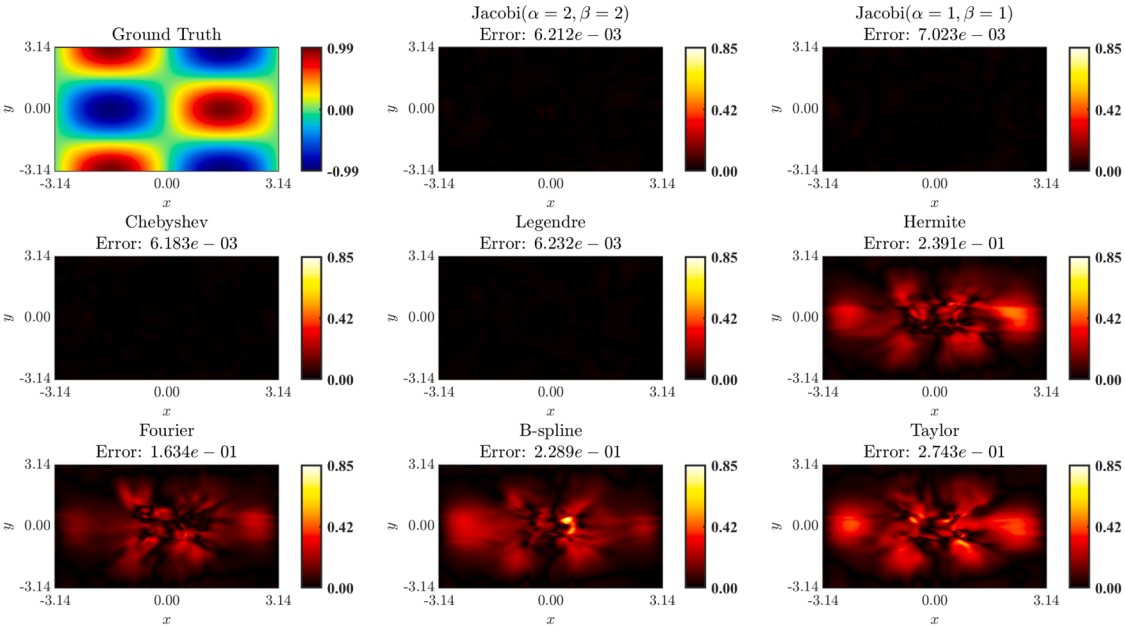


Fig. 11. Comparison of different physics-informed models for Taylor–Green vortex equation prediction on u: Ground truth vs. different physics-informed models with their respective relative l_2 errors.

Fig. 14 shows training convergence for different basis functions. Chebyshev models demonstrate substantial advantages: rapid initial convergence (0–500 iterations) with order-of-magnitude improvement over Hermite, Fourier, and Taylor functions; stable mid-phase convergence (500–2000 iterations) with test error decreasing to 10^{-3} range; final convergence achieving 4.1×10^{-3} test error—115× and 58× better than B-spline (4.7×10^{-1}) and Taylor (2.4×10^{-1}), respectively. This performance stems from Jacobi basis function orthogonality, enabling effective multi-scale feature capture in high-dimensional flows.

3.4. Kovasznay flow

we demonstrate the performance of J-PIKAN in solving steady-state flow field equations using the Kovasznay flow problem. The corresponding 2D steady-state Navier–Stokes equations and their analytical solutions are as follows:

$$\begin{aligned} u(x, y) &= 1 - e^{\zeta x} \cos(2\pi y), \\ v(x, y) &= \frac{\zeta}{2\pi} e^{\zeta x} \sin(2\pi y), \\ p(x, y) &= \frac{1}{2}\left(1 - e^{2\zeta x}\right), \end{aligned} \tag{3.11}$$

where

$$\zeta = \frac{1}{2\nu} - \sqrt{\frac{1}{4\nu^2} + 4\pi^2}, \quad \nu = \frac{1}{\mathrm{Re}} = \frac{1}{40}, \tag{3.12}$$

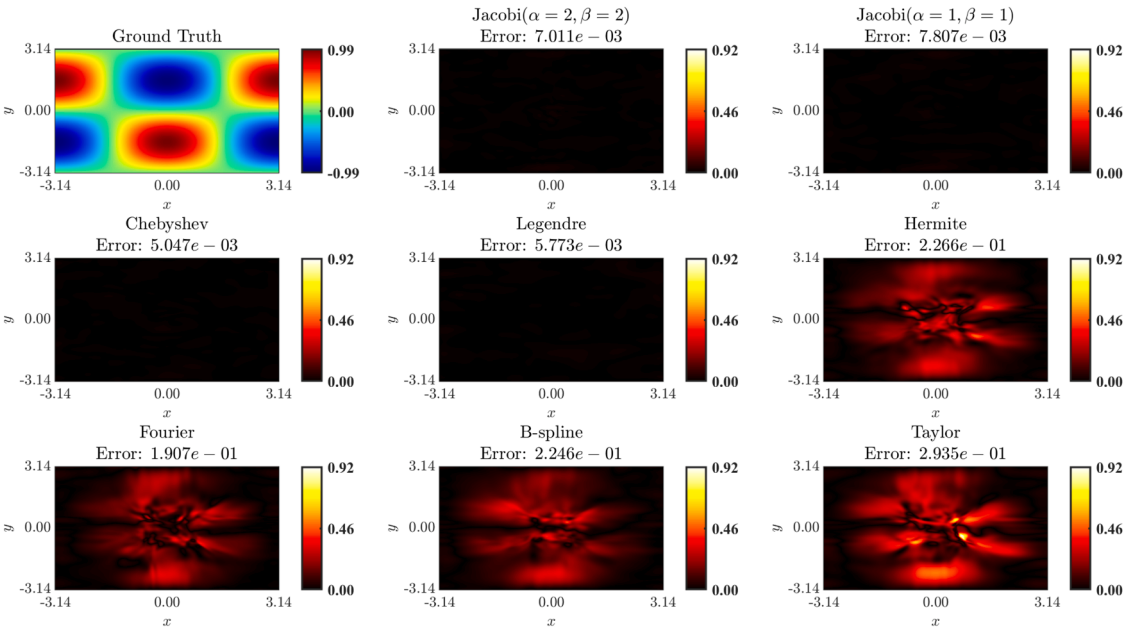


Fig. 12. Comparison of different physics-informed models for Taylor–Green vortex equation prediction on v: Ground truth vs. different physics-informed models with their respective relative l_2 errors.

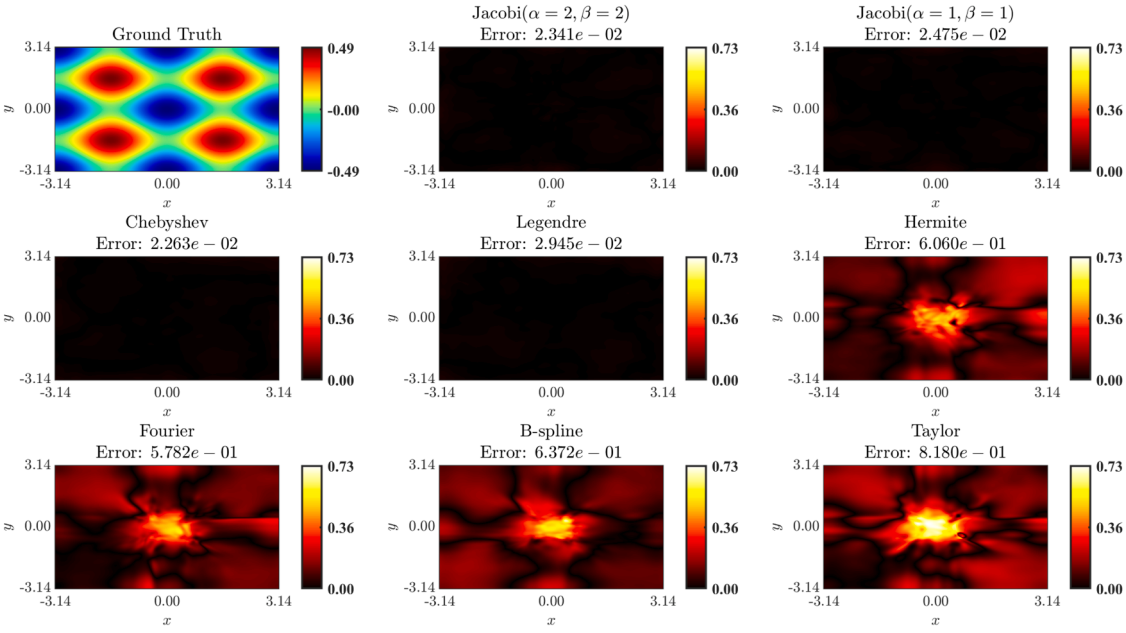


Fig. 13. Comparison of different physics-informed models for Taylor–Green vortex equation prediction on p: Ground truth vs. different physics-informed models with their respective relative l_2 errors.

We consider the computational domain as $[-0.5, 1.0] \times [-0.5, 1.5]$. For the boundary conditions, we sample 101 fixed spatial points along each boundary, resulting in a total of $N_b = 4 \times 101$. To compute the equation loss, we randomly select 2000 points within the domain. The steady-state flow has no initial conditions. In this case, we only use the L-BFGS optimizer with a maximum of 10,000 optimization iterations.

To systematically evaluate different basis functions in physics-informed neural networks, we conducted experiments on the Kovasznay flow using unified architecture [2, 30, 30, 30, 30, 3] with identical parameters (14,250) across all models for fair comparison.

Table 8 shows prediction errors for different basis functions. Jacobi polynomials ($\alpha = \beta = 2$) achieved optimal performance across all quantities: u-component error of 1.5×10^{-3}, v-component error of 8.2×10^{-3}, and pressure error of 7.8×10^{-3}. Jacobi ($\alpha = \beta = 1$)

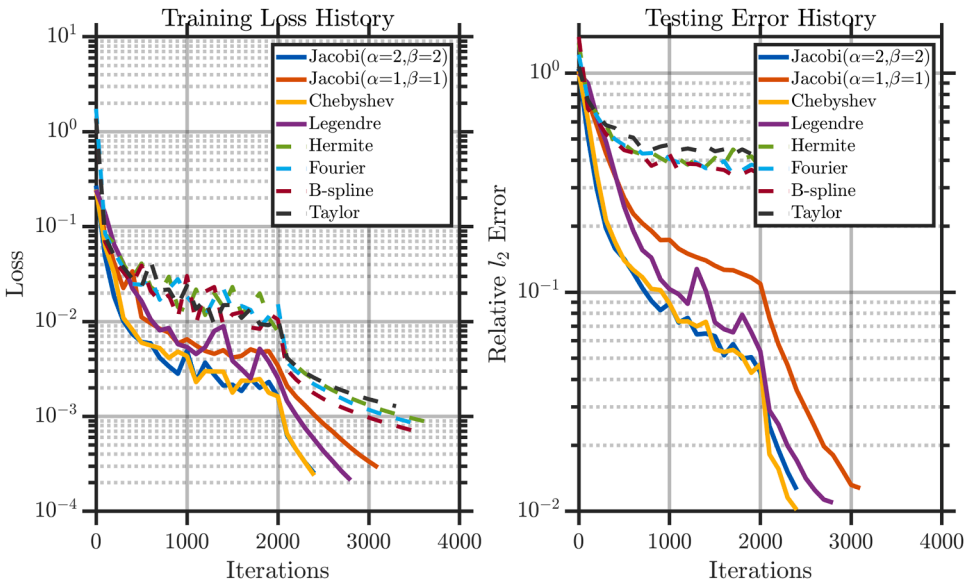


Fig. 14. Training loss and testing error histories for the Taylor–Green vortex equation: Convergence comparison among different basis functions including Jacobi polynomials ($\alpha = \beta = 2, 1, 0, -0.5$), and some classical orthogonal polynomials.

Table 8
Kovasznay flow computational setup (Re = 40): steady-state domain $[-0.5, 1.0] \times [-0.5, 1.5]$, boundary discretization with 101 points per edge (404 total), 2000 randomly sampled interior points, L-BFGS-only optimization (maximum 10,000 iterations), identical architecture [2,30,30,30,30,3] with 14,250 parameters for fair comparison across eight basis function types, multi-component validation framework, evaluated over 5 random seeds.

Model	Network Architecture	N. Params	Rel. l_2 Error u	Rel. l_2 Error v	Rel. l_2 Error p
Jacobi($\alpha = \beta = 2$)	[2, 30, 30, 30, 30, 3]	14,250	$(1.5 \pm 0.4) \times 10^{-3}$	$(8.2 \pm 2.3) \times 10^{-3}$	$(7.8 \pm 2.1) \times 10^{-3}$
Jacobi($\alpha = \beta = 1$)	[2, 30, 30, 30, 30, 3]	14,250	$(1.9 \pm 0.5) \times 10^{-3}$	$(9.3 \pm 2.5) \times 10^{-3}$	$(8.5 \pm 2.4) \times 10^{-3}$
Chebyshev	[2, 30, 30, 30, 30, 3]	14,250	$(2.7 \pm 0.7) \times 10^{-3}$	$(9.8 \pm 2.9) \times 10^{-3}$	$(8.1 \pm 2.2) \times 10^{-3}$
Legendre	[2, 30, 30, 30, 30, 3]	14,250	$(6.5 \pm 2.2) \times 10^{-1}$	$(1.0 \pm 0.3) \times 10^{0}$	$(9.8 \pm 3.1) \times 10^{-1}$
Hermite	[2, 30, 30, 30, 30, 3]	14,250	$(1.9 \pm 0.6) \times 10^{-3}$	$(9.3 \pm 2.7) \times 10^{-3}$	$(4.2 \pm 1.3) \times 10^{-3}$
Bessel	[2, 30, 30, 30, 30, 3]	14,250	$(7.5 \pm 2.1) \times 10^{-3}$	$(4.4 \pm 1.3) \times 10^{-2}$	$(1.9 \pm 0.6) \times 10^{-2}$
Laguerre	[2, 30, 30, 30, 30, 3]	14,250	$(3.1 \pm 0.9) \times 10^{-3}$	$(2.3 \pm 0.7) \times 10^{-2}$	$(9.7 \pm 2.8) \times 10^{-3}$
Taylor	[2, 30, 30, 30, 30, 3]	14,250	$(6.6 \pm 2.3) \times 10^{-1}$	$(1.7 \pm 0.6) \times 10^{0}$	$(8.7 \pm 2.9) \times 10^{-1}$

and Chebyshev models also demonstrated high accuracy, indicating that orthogonal polynomial basis functions with appropriate parameters effectively capture flow physics. In contrast, Taylor and Legendre models showed poor performance with errors ranging from 10^{-1} to 10^{0}. Notably, Hermite polynomials performed well for pressure prediction (4.2×10^{-3}), suggesting basis-specific advantages for certain quantities.

Figs. 15–17 demonstrate Jacobi models' superior flow structure capture. The $\alpha = \beta = 2$ configuration shows highest fidelity in recovering complex flow features, particularly in steep gradient regions like boundary layers and pressure gradients.

Fig. 18 reveals training dynamics differences. While all models show rapid initial loss decrease, Jacobi polynomials ($\alpha = \beta = 2$) exhibit optimal convergence: achieving low loss early and maintaining steady improvement to the lowest final error. This performance stems from Jacobi polynomials' orthogonal properties enabling effective function space approximation. The $\alpha = \beta = 2$ configuration demonstrates clear advantages over $\alpha = \beta = 1$ in both convergence speed and accuracy, highlighting the importance of hyperparameter selection for optimal J-PIKAN configuration.

3.5. Flow in a lid-driven cavity

Finally, to assess the performance of J-PIKAN in solving complex flow field problems at high Reynolds numbers, we choose a challenging problem for MLP-PINNs: the lid-driven cavity flow at high Reynolds numbers ($Re = 1000$, $Re = 2500$ and $Re = 4000$). The lid-driven cavity flow at high large numbers exhibits complex secondary vortices near the corners [58], posing significant challenges to traditional PINNs.This is a classic benchmark problem in computational fluid dynamics (CFD), governed by the two-dimensional incompressible Navier–Stokes equations:

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} - \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0$$

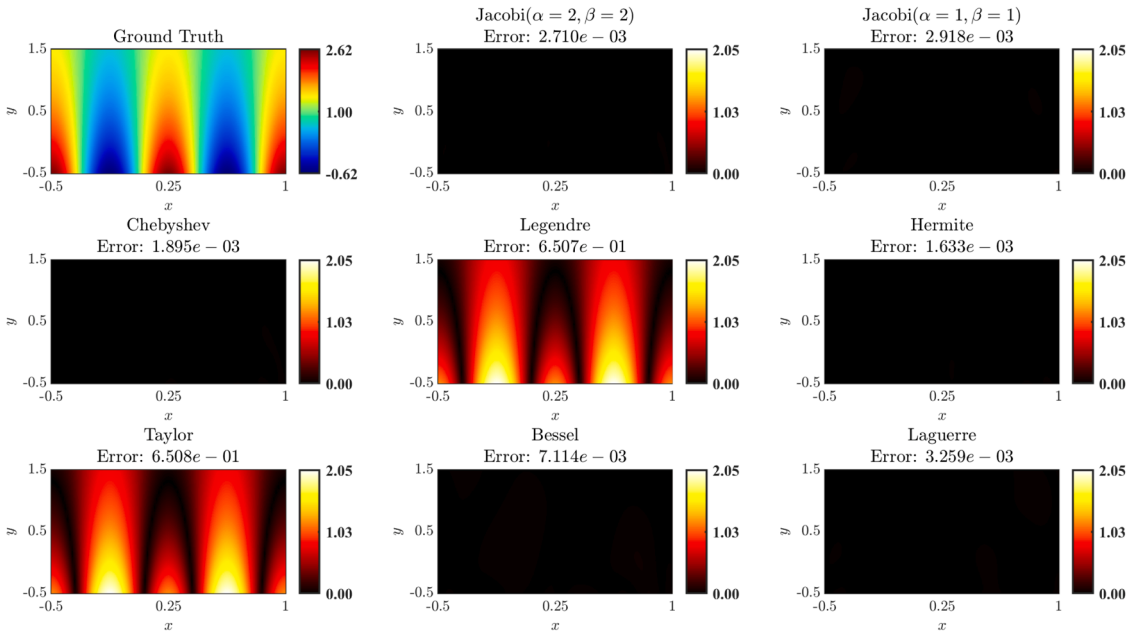


Fig. 15. Comparison of different physics-informed models for Kovasznay flow equation prediction on u: Ground truth vs. different physics-informed models with their respective relative l_2 errors.

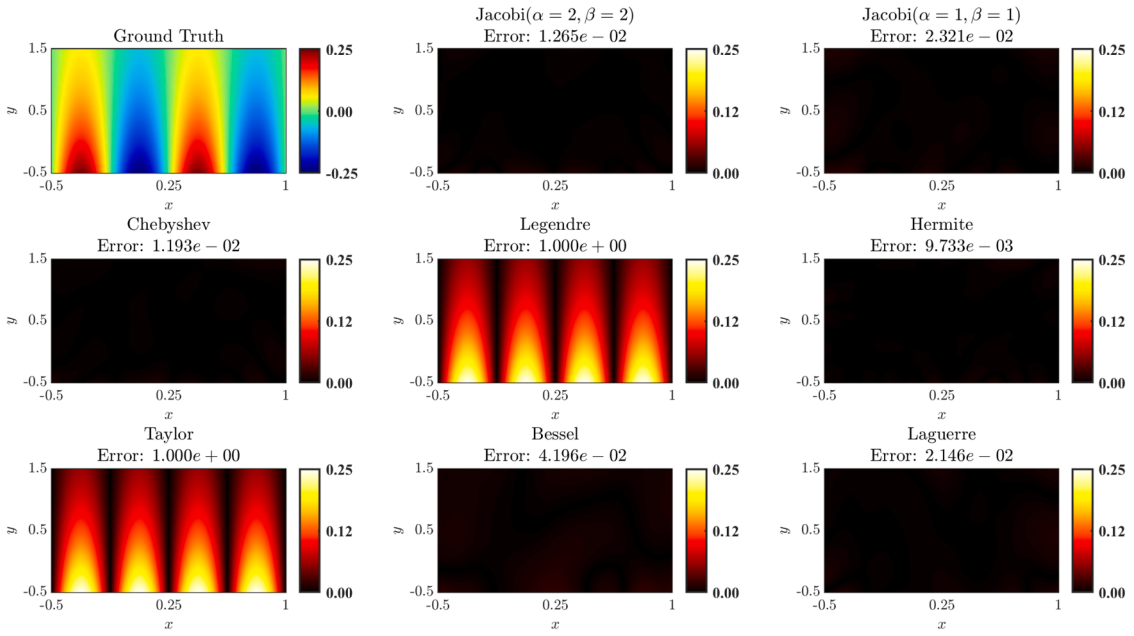


Fig. 16. Comparison of different physics-informed models for Kovasznay flow equation prediction on v: Ground truth vs. different physics-informed models with their respective relative l_2 errors.

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} - \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) = 0$$
$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{3.13}$$

Here, $\boldsymbol{u} = (u, v)$ is the velocity vector, and p is the pressure. The computational domain $\Omega = (0, 1) \times (0, 1)$ is a two-dimensional cavity, with the velocity at the top boundary Γ_0 set to $(u, v) = (1, 0)$, and $(u, v) = (0, 0)$ on the other three boundaries Γ_1. Despite its simple geometry, the flow in the cavity exhibits rich fluid physical characteristics, such as multiple counter-rotating recirculation regions,

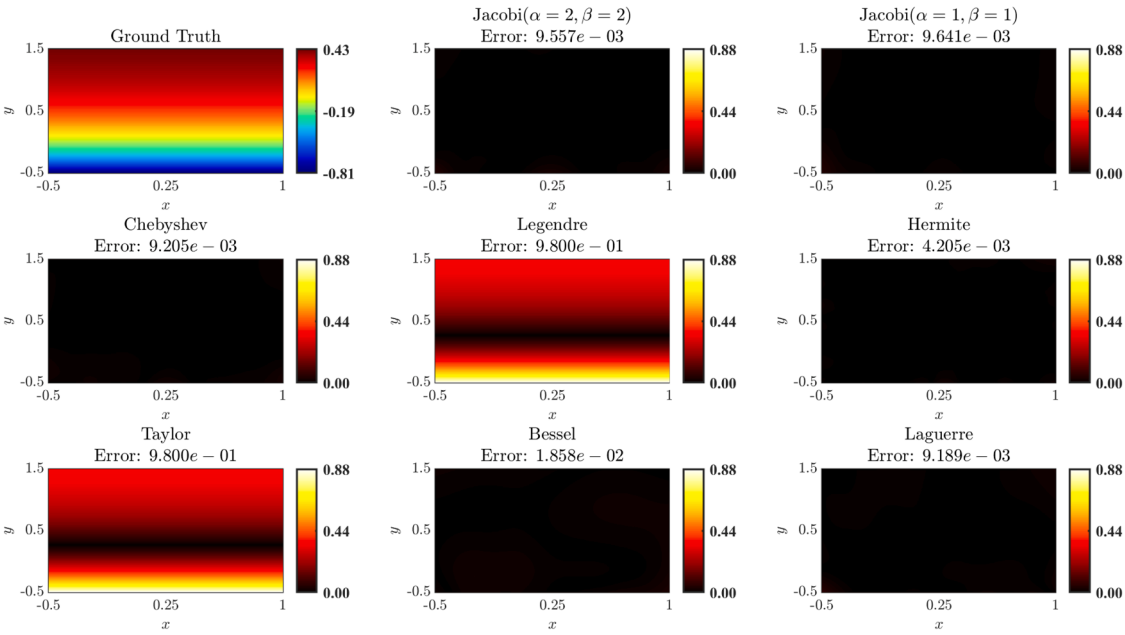


Fig. 17. Comparison of different physics-informed models for Kovasznay flow equation prediction on p: Ground truth vs. different physics-informed models with their respective relative l_2 errors.

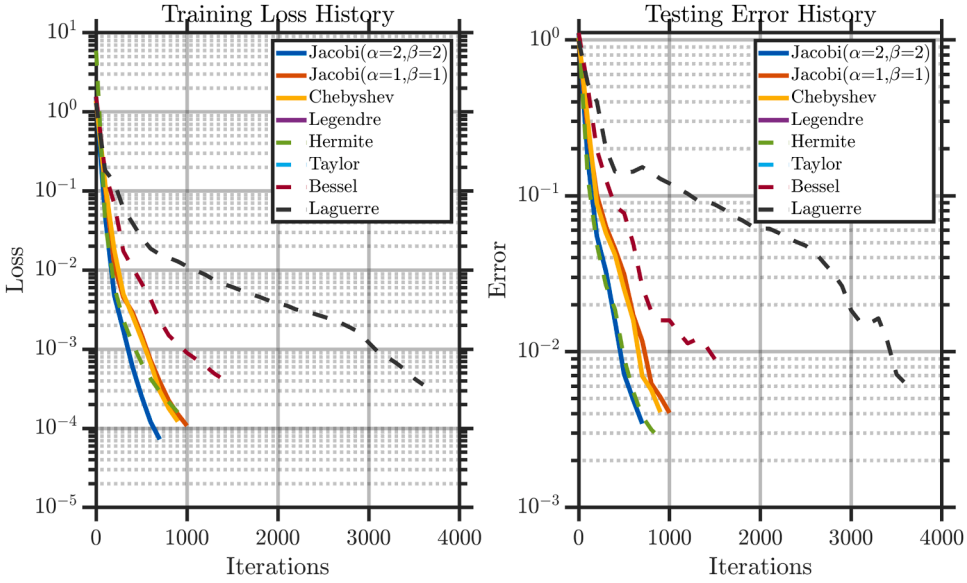


Fig. 18. Training loss and testing error histories for the Kovasznay flow equation: Convergence comparison among different basis functions including Jacobi polynomials ($\alpha = \beta = 2$, $\alpha = \beta = 1$), and some classical orthogonal polynomials.

as the Reynolds number increases. Studies have shown that when Re > 100, standard MLP-PINNs are unable to solve this problem, and the difficulty of solving it increases as the Reynolds number grows [30].

The solution of lid-driven cavity flows at high Reynolds numbers presents significant numerical challenges, primarily due to the severe ill-conditioning inherent in MLP-PINNs. To address this computational obstacle, we propose a physics-informed neural network framework incorporating pseudo-time stepping [32,33]. This methodology advances the solution from arbitrary initial conditions to steady-state through pseudo-time evolution under specified boundary conditions. The implicit time-stepping equation governing this process is formulated as:

$$\bar{u}_{n+1} = \hat{u}_n + \Delta\tau \cdot \mathcal{N}[\bar{u}_{n+1}], \tag{3.14}$$

Table 9
Performance comparison of different physics-informed models for solving the lid-driven cavity equation at $Re = 1000$ (results from 5 random seeds).

Model	Network Architecture	N. Params	Relative l_2 Error
Jacobi($\alpha = \beta = 2$)	[2, 20, 20, 20, 3]	4500	$(4.2 \pm 1.3) \times 10^{-2}$
Jacobi($\alpha = \beta = 1$)	[2, 20, 20, 20, 3]	4500	$(4.1 \pm 1.1) \times 10^{-2}$
Chebyshev	[2, 20, 20, 20, 3]	4500	$(4.4 \pm 1.2) \times 10^{-2}$
MLP	[2, 50, 50, 50, 50, 50, 3]	10,503	$(2.2 \pm 0.8) \times 10^{-1}$

Table 10
Performance analysis of neural networks for lid-driven cavity flow at $Re = 1000$: Impact of polynomial degree on accuracy with fixed width and depth (results from 5 random seeds).

Degree	Width	Depth	N. Params	Relative l_2 Error
2	20	4	3780	$(9.6 \pm 2.8) \times 10^{-2}$
3	20	4	5040	$(5.3 \pm 1.6) \times 10^{-2}$
4	20	4	6300	$(4.7 \pm 1.3) \times 10^{-2}$

Table 11
Comparative analysis of neural networks for lid-driven cavity flow at $Re = 2500$: Performance and parameter efficiency of Jacobi-based models versus traditional MLPs (results from 5 random seeds).

Model	Network Architecture	N. Params	Relative l_2 Error
Jacobi($\alpha = \beta = 2$)	[2, 20, 20, 20, 3]	4500	$(6.0 \pm 1.7) \times 10^{-2}$
Jacobi($\alpha = \beta = 1$)	[2, 20, 20, 20, 3]	4500	$(5.5 \pm 1.4) \times 10^{-2}$
Chebyshev	[2, 20, 20, 20, 3]	4500	$(7.0 \pm 1.9) \times 10^{-2}$
MLP	[2, 50, 50, 50, 50, 50, 3]	10,503	$(1.7 \pm 0.6) \times 10^{-1}$

where $\mathcal{N}[\cdot]$ denotes the operator involved in the time-stepping process, $\hat{u}_n$ represents the neural network at the nth optimization step, with parameters θ_n. For comprehensive evaluation and fair comparison, this pseudo-time stepping strategy is consistently implemented across all neural network models examined in lid-driven cavity flows.

To investigate physics-informed neural network performance in high Reynolds number flows, we examine lid-driven cavity flow at $Re = 1000$, 2500, and 4000, systematically comparing basis functions and architectures.

Table 9 compares models for $Re = 1000$ using architecture [2, 20, 20, 20, 3] with 4500 parameters for orthogonal polynomial models. Jacobi polynomials ($\alpha = \beta = 2, 1$) achieve relative L_2 errors of approximately 4.2×10^{-2}, comparable to Chebyshev (4.4×10^{-2}). Despite using deeper architecture [2, 50, 50, 50, 50, 50, 3] with twice the parameters, traditional MLPs achieve higher error (2.2×10^{-1})—four times worse than J-PIKAN.

Table 10 examines polynomial degree effects with fixed width (20) and depth (4). Increasing degree from 2 to 4 reduces error significantly from 9.6×10^{-2} to 4.7×10^{-2} with moderate parameter increase (3780 to 6300). The largest improvement occurs from degree 2 to 3, suggesting optimal cost-accuracy balance.

Fig. 20 shows that orthogonal polynomial models exhibit superior training dynamics. J-PIKAN ($\alpha = \beta = 2$) demonstrates fastest convergence and most stable behavior, effectively satisfying governing equation constraints and boundary conditions—a key advantage over MLPs.

For $Re = 2500$ (Table 11), Jacobi polynomials ($\alpha = \beta = 1$) achieve 5.5×10^{-2} error versus Chebyshev's 7.0×10^{-2}. MLPs with 10,503 parameters (nearly double) fail with 1.7×10^{-1} error. Fig. 21 confirms superior convergence and stability for orthogonal polynomial models.

At $Re = 4000$ (Fig. 22), J-PIKAN [2,30,30,30,30,4] with 14,250 parameters achieves 6.8×10^{-2} error, while MLP [2,100,100,100,100,4] with 21,106 parameters yields 1.8×10^{-1} error. This represents 32.5 % parameter reduction with 62.2 % accuracy improvement, demonstrating superior computational efficiency for high Reynolds number flows.

Fig. 19 shows a comparison of different physics-informed models for predicting $U = \sqrt{u^2 + v^2}$ at $Re = 1000$, comparing the ground truth with the predictions from different models.

3.6. Computational efficiency and sensitivity analysis

To comprehensively evaluate J-PIKAN's practical applicability, we conduct detailed analyses of computational efficiency and hyperparameter sensitivity. These investigations provide crucial insights for practitioners seeking to implement J-PIKAN in real-world fluid dynamics applications.

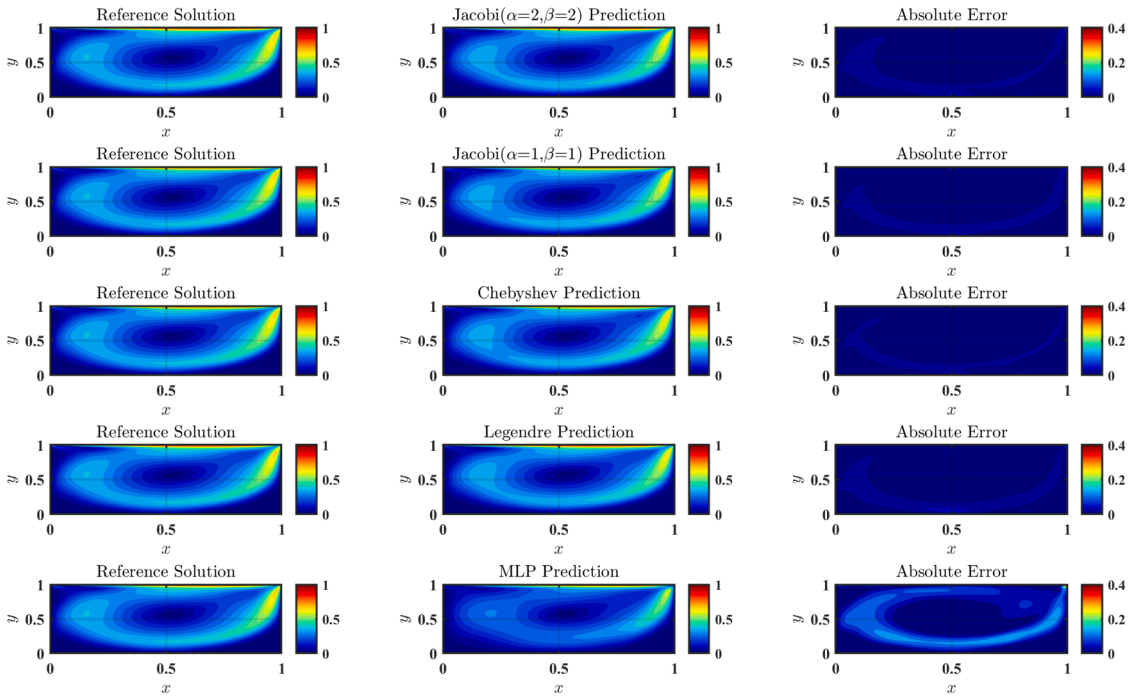


Fig. 19. Comparison of different physics-informed models for lid-driven cavity flow equation prediction on U when Re = 1000: Ground truth vs. different models.

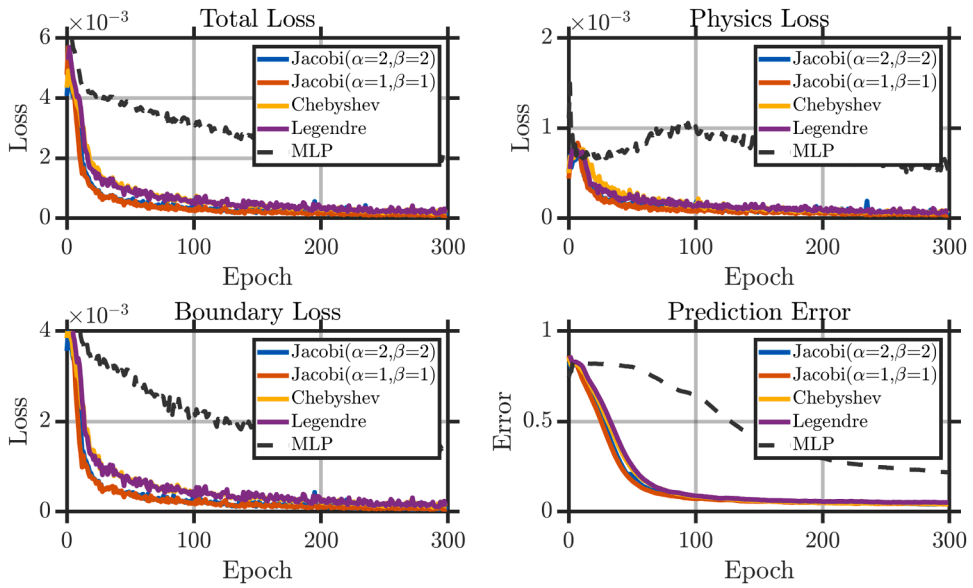


Fig. 20. Training dynamics comparison for lid-driven cavity flow when Re = 1000: Evolution of total loss, physics loss, boundary loss, and relative error across different models.

3.6.1. Training time analysis

Fig. 23 presents a comprehensive comparison of training times across different basis functions for four representative fluid dynamics problems. The analysis reveals several important computational characteristics of J-PIKAN compared to alternative approaches.

For the Burgers equation, J-PIKAN demonstrates competitive computational efficiency with training times ranging from 760 to 930 s across different Jacobi polynomial configurations ($\alpha = \beta = 2$ and $\alpha = \beta = 1$). Notably, these times are comparable to or faster than other orthogonal polynomial approaches such as Chebyshev (1051s) and Legendre (791s), while significantly outperforming

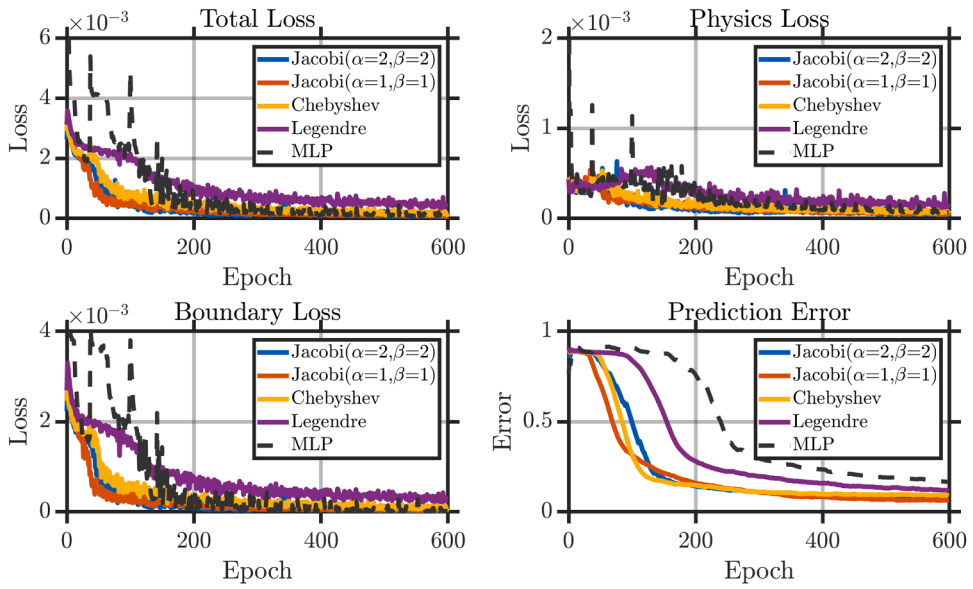


Fig. 21. Training dynamics comparison for lid-driven cavity flow when $Re = 2500$: Evolution of total loss, physics loss, boundary loss, and relative error across different physics-informed models.

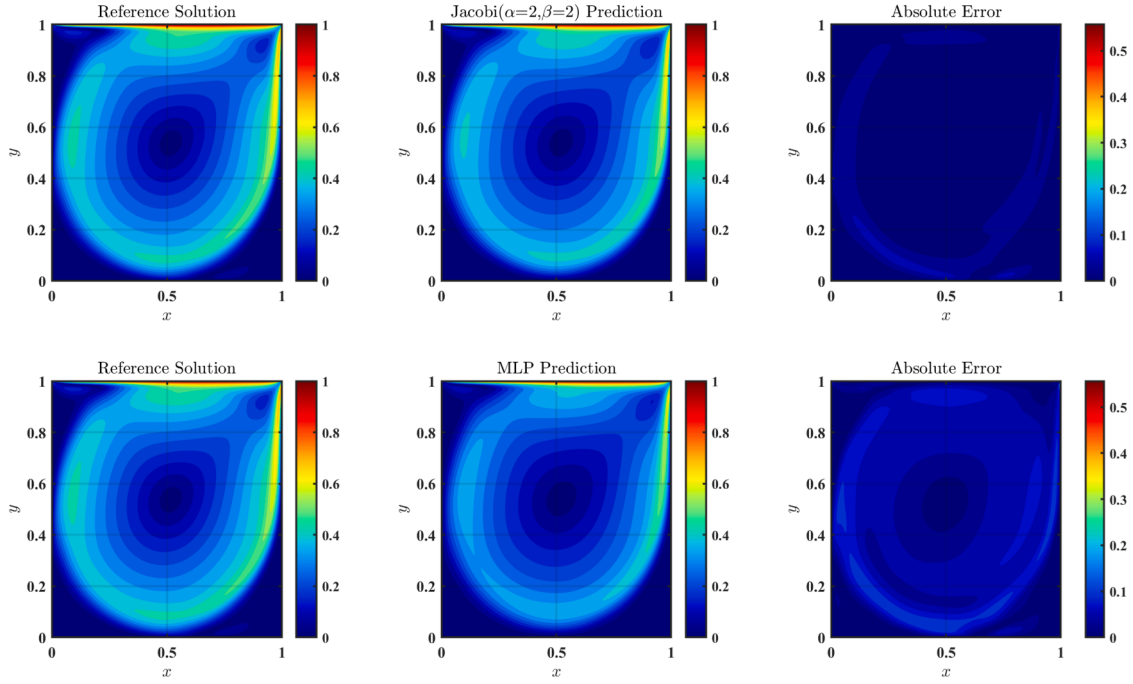


Fig. 22. Comparison of different physics-informed models for lid-driven cavity flow equation prediction on U when $Re = 4000$: Ground truth vs. different models.

traditional MLPs (291s). Although MLPs exhibit the fastest training time, this advantage is negated by their substantially higher error rates, as demonstrated in previous sections.

The KdV equation results show more pronounced differences in computational overhead. J-PIKAN requires 2680–3021 s for convergence, which is comparable to other polynomial-based methods (Chebyshev: 2915s, Legendre: 3051s) but substantially higher than MLPs (940s). However, this increased computational cost is justified by the significant accuracy improvements achieved by J-PIKAN. The higher computational demand for KdV problems can be attributed to the third-order derivatives requiring more complex automatic differentiation operations.

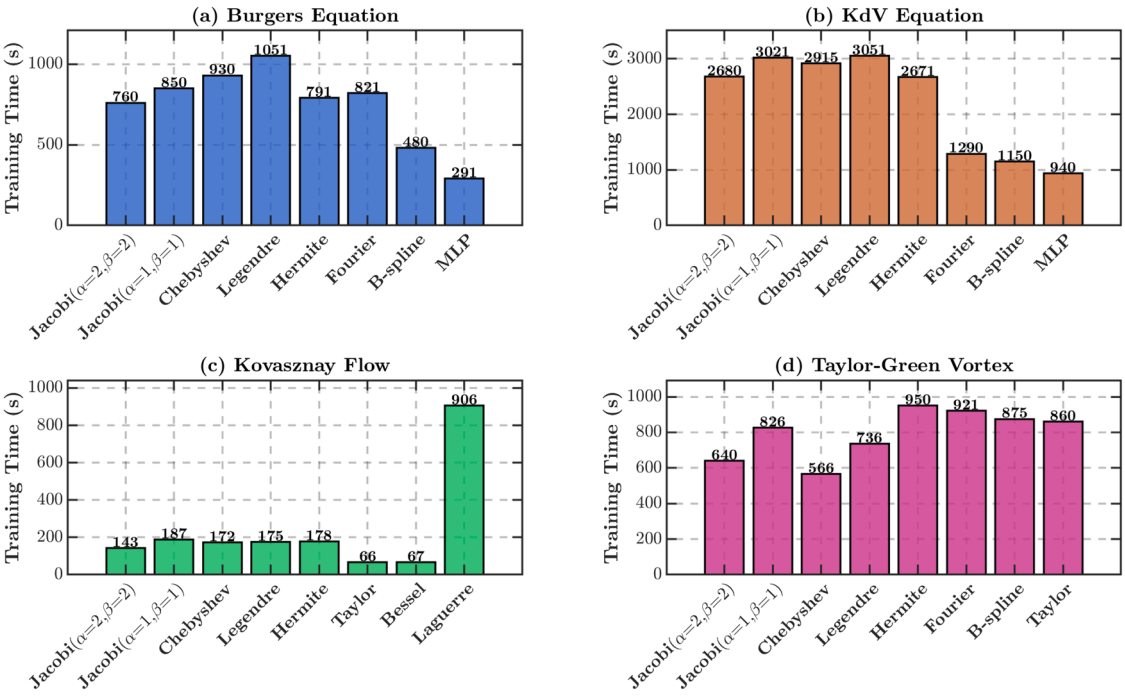


Fig. 23. Training time comparison across different basis functions for four fluid dynamics benchmark problems: (a) Burgers equation, (b) KdV equation, (c) Kovasznay flow, and (d) Taylor–Green vortex. The results demonstrate J-PIKAN's computational efficiency relative to other polynomial-based methods and traditional MLPs.

Table 12
Sensitivity analysis of polynomial degree on solution accuracy for J-PIKAN with Jacobi polynomials ($\alpha = \beta = 2$). For NS equations, values represent average error across u, v, and p components (results from 5 random seeds).

Polynomial Degree	Burgers	KdV	Kovasznay Flow	Taylor–Green Vortex
2	$(2.3 \pm 0.6) \times 10^{-3}$	$(1.8 \pm 0.5) \times 10^{-2}$	$(4.83 \pm 1.35) \times 10^{-2}$	$(2.90 \pm 0.87) \times 10^{-2}$
4	$(5.8 \pm 1.7) \times 10^{-4}$	$(8.6 \pm 2.4) \times 10^{-3}$	$(5.83 \pm 1.63) \times 10^{-3}$	$(7.13 \pm 2.14) \times 10^{-3}$
6	$(6.2 \pm 1.9) \times 10^{-4}$	$(9.1 \pm 2.5) \times 10^{-3}$	$(6.15 \pm 1.72) \times 10^{-3}$	$(7.58 \pm 2.27) \times 10^{-3}$
8	$(7.1 \pm 2.0) \times 10^{-4}$	$(9.8 \pm 2.7) \times 10^{-3}$	$(6.92 \pm 1.94) \times 10^{-3}$	$(8.23 \pm 2.47) \times 10^{-3}$

For steady-state problems like Kovasznay flow, J-PIKAN demonstrates excellent computational efficiency with training times of 143–187 s, significantly outperforming MLPs (906s). This efficiency stems from the superior convergence properties of orthogonal polynomials in steady-state optimization landscapes, where the well-conditioned nature of the Jacobi basis functions facilitates faster convergence to optimal solutions.

The Taylor–Green vortex results show moderate computational overhead for J-PIKAN (640-826s) compared to other polynomial methods, with all approaches requiring similar training times except for the notably efficient Chebyshev implementation (566s). The computational cost remains reasonable for this multi-physics problem involving coupled velocity and pressure fields.

3.6.2. Sensitivity analysis

Polynomial degree sensitivity. Table 12 examines the impact of polynomial degree on solution accuracy across four benchmark problems. The analysis reveals consistent patterns in the degree-accuracy relationship for J-PIKAN with Jacobi polynomials ($\alpha = \beta = 2$).

The results demonstrate a clear trend: increasing polynomial degree from 2 to 4 yields substantial accuracy improvements across all test cases. For the Burgers equation, the error decreases dramatically from 2.3×10^{-3} to 5.8×10^{-4}—a four-fold improvement. Similarly, the KdV equation shows a two-fold error reduction from 1.8×10^{-2} to 8.6×10^{-3}. The Kovasznay flow exhibits the most dramatic improvement, with error decreasing by nearly an order of magnitude from 4.83×10^{-2} to 5.83×10^{-3}.

However, the benefits of increasing polynomial degree beyond 4 show diminishing returns. For degrees 6 and 8, the error improvements are marginal and sometimes exhibit slight degradation, particularly evident in the KdV and Taylor–Green vortex cases. This behavior suggests that degree 4 represents an optimal balance between expressiveness and overfitting for the tested problems. The slight performance degradation at higher degrees may be attributed to increased model complexity leading to overfitting on the finite training data, despite the orthogonality properties of Jacobi polynomials.

Table 13
Sensitivity analysis of learning rate schedules on solution accuracy for J-PIKAN across different fluid dynamics problems. For NS equations, values represent average error across u, v, and p components (results from 5 random seeds).

Learning Rate Schedule	Burgers	KdV	Kovasznay Flow	Taylor–Green Vortex
Constant (1e-3)	$(1.8 \pm 0.5) \times 10^{-3}$	$(1.2 \pm 0.4) \times 10^{-2}$	$(1.17 \pm 0.35) \times 10^{-2}$	$(1.97 \pm 0.59) \times 10^{-2}$
Exponential Decay	$(1.2 \pm 0.3) \times 10^{-3}$	$(9.8 \pm 2.7) \times 10^{-3}$	$(8.33 \pm 2.33) \times 10^{-3}$	$(1.57 \pm 0.47) \times 10^{-2}$
Cosine Annealing	$(5.8 \pm 1.7) \times 10^{-4}$	$(8.6 \pm 2.4) \times 10^{-3}$	$(5.83 \pm 1.63) \times 10^{-3}$	$(7.13 \pm 2.14) \times 10^{-3}$
Step Decay	$(9.7 \pm 2.4) \times 10^{-4}$	$(1.0 \pm 0.3) \times 10^{-2}$	$(7.20 \pm 2.02) \times 10^{-3}$	$(1.21 \pm 0.36) \times 10^{-2}$
Warm Restart	$(7.2 \pm 1.8) \times 10^{-4}$	$(9.2 \pm 2.5) \times 10^{-3}$	$(6.70 \pm 1.88) \times 10^{-3}$	$(9.77 \pm 2.93) \times 10^{-3}$

These findings provide practical guidance for practitioners: polynomial degree 4 emerges as a robust choice across diverse fluid dynamics problems, offering significant accuracy improvements over lower degrees while avoiding the potential overfitting issues associated with higher degrees.

Learning rate schedule sensitivity. Table 13 investigates the impact of different learning rate schedules on J-PIKAN's convergence and final accuracy. This analysis is crucial for understanding the optimization dynamics of polynomial-based physics-informed neural networks.

The results reveal that cosine annealing consistently achieves the best performance across all test problems. For the Burgers equation, cosine annealing achieves the lowest error of 5.8×10^{-4}, representing a significant improvement over the constant learning rate baseline (1.8×10^{-3}). This superior performance is maintained across all problems: KdV equation (8.6×10^{-3} vs. 1.2×10^{-2}), Kovasznay flow (5.83×10^{-3} vs. 1.17×10^{-2}), and Taylor–Green vortex (7.13×10^{-3} vs. 1.97×10^{-2}).

The success of cosine annealing can be attributed to its ability to provide both exploration and exploitation phases during training. The periodic restart mechanism allows the optimizer to escape local minima while the gradual learning rate reduction enables fine-tuning near optimal solutions. This is particularly beneficial for physics-informed neural networks, where the loss landscape can be complex due to the interplay between PDE residuals, boundary conditions, and initial conditions.

Exponential decay shows consistent improvements over constant learning rates but generally underperforms compared to cosine annealing. Step decay demonstrates moderate performance, while warm restart shows promise, particularly for the Burgers equation where it achieves the second-best performance (7.2×10^{-4}).

The sensitivity analysis provides several practical recommendations for J-PIKAN implementation: (1) polynomial degree 4 offers optimal accuracy-complexity trade-offs for most fluid dynamics problems; (2) cosine annealing learning rate schedules should be preferred for optimal convergence; (3) the method exhibits robust performance across different configurations, indicating practical reliability for diverse applications.

These findings establish J-PIKAN as a computationally efficient and robust approach for solving fluid dynamics problems, with clear guidelines for optimal hyperparameter selection.

4. Conclusion

This work addresses the fundamental limitations of traditional MLP-based Physics-Informed Neural Networks by developing J-PIKAN, a physics-informed Kolmogorov–Arnold Network based on Jacobi orthogonal polynomials. Through systematic comparison of different basis functions across five representative fluid dynamics benchmarks, we demonstrate that Jacobi polynomials consistently achieve superior performance, delivering 1–2 orders of magnitude improvement in solution accuracy compared to baseline MLPs while requiring only 50 % of the parameters. The orthogonality of Jacobi polynomials significantly reduces numerical ill-conditioning during training, as evidenced by Hessian eigenvalue analysis, enabling successful capture of complex phenomena including shock wave propagation, soliton dynamics, and high Reynolds number flows ($Re = 4000$). These achievements position J-PIKAN as a promising framework for developing deep learning-based solvers for fluid dynamics applications.

However, several limitations warrant future investigation, particularly the significant memory consumption and computational overhead introduced by high-order polynomial implementations, as well as training stability issues that occasionally emerge with high-order polynomials in problems with sharp gradients or discontinuous solutions. To address these challenges, future research should focus on developing more efficient and stable network architectures and optimization algorithms specifically tailored for polynomial-based physics-informed neural networks. Variable separation architectures could significantly reduce computational complexity by decomposing high-dimensional problems into lower-dimensional subproblems, while adaptive polynomial degree selection and hybrid approaches may enhance both stability and efficiency, further advancing J-PIKAN as a powerful tool for solving complex fluid dynamics problems

CRediT authorship contribution statement

Xiong Xiong: Writing – original draft, Methodology, Conceptualization; **Kang Lu:** Investigation, Formal analysis, Data curation; **Zhuo Zhang:** Visualization, Software; **Zheng Zeng:** Validation, Resources, Formal analysis; **Sheng Zhou:** Validation, Data curation; **Zichen Deng:** Methodology, Investigation; **Rongchun Hu:** Writing – review & editing, Validation, Data curation.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Natural Science Foundation of China through the grant nos. 11972293, 12232015, and KGJ national defense technology foundation: JSZL2024607B001.

References

[1] Evans LC. Partial differential equations. American Mathematical Society; 2022.
[2] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.
[3] Cai S, Mao Z, Wang Z, Yin M, Karniadakis GE. Physics-informed neural networks (PINNs) for fluid mechanics: a review. Acta Mech Sin 2021;37(12):1727–38.
[4] Arthurs CJ, King AP. Active training of physics-informed neural networks to aggregate and interpolate parametric solutions to the Navier–Stokes equations. J Comput Phys 2021;438:110364.
[5] Jin X, Cai S, Li H, Karniadakis GE. NSFnets (Navier–Stokes flow nets): physics-informed neural networks for the incompressible Navier–Stokes equations. J Comput Phys 2021;426:109951.
[6] Eivazi H, Tahani M, Schlatter P, Vinuesa R. Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations. Phys Fluids 2022;34(7):075117.
[7] Gu L, Qin S, Xu L, Chen R. Physics-informed neural networks with domain decomposition for the incompressible Navier–Stokes equations. Phys Fluids 2024;36(2):021914.
[8] Wang J, Xiao X, Feng X, Xu H. An improved physics-informed neural network with adaptive weighting and mixed differentiation for solving the incompressible Navier–Stokes equations. Nonlinear Dyn 2024;112(18):16113–34.
[9] Bounnah Y, Mihoubi MK, Larbi S. Physics informed neural network with Fourier feature for natural convection problems. Eng Appl Artif Intell 2025;146:110327.
[10] Botarelli T, Fanfani M, Nesi P, Pinelli L. Using physics-informed neural networks for solving Navier–Stokes equations in fluid dynamic complex scenarios. Eng Appl Artif Intell 2025;148:110347.
[11] Xiong X, Lu K, Zhang Z, Zeng Z, Zhou S, Hu R, et al. High-frequency flow field super-resolution via physics-informed hierarchical adaptive fourier feature networks. Phys Fluids 2025;37(9):097111.
[12] Xiong X, Zhang Z, Hu R, Gao C, Deng Z. Separated-variable spectral neural networks: a physics-informed learning approach for high-frequency PDEs. 2025. arXiv:2508.00628.
[13] Chen Z, Liu Y, Sun H. Physics-informed learning of governing equations from scarce data. Nat Commun 2021;12(1):6136.
[14] McGreivy N, Hakim A. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. Nat Mach Intell 2024;6(10):1256–69.
[15] Brunton SL, Kutz JN. Promising directions of machine learning for partial differential equations. Nat Comput Sci 2024;4(7):483–94.
[16] Hu H, Qi L, Chao X. Physics-informed neural networks (PINN) for computational solid mechanics: numerical frameworks and applications. Thin-Walled Struct 2024;205:112495.
[17] Diao Y, Yang J, Zhang Y, Zhang D, Du Y. Solving multi-material problems in solid mechanics using physics-informed neural networks based on domain decomposition technology. Comput Methods Appl Mech Eng 2023;413:116120.
[18] Söyleyici C, Ünver HÖ. A physics-informed deep neural network based beam vibration framework for simulation and parameter identification. Eng Appl Artif Intell 2025;141:109804.
[19] Jin Z, Hou S, Zhang H, Jia W. Radial-basis-function neural network for solving the transient probability density function under composite stochastic noise. Phys Rev E 2025;112(1):015306.
[20] Zhai W, Tao D, Bao Y. Parameter estimation and modeling of nonlinear dynamical systems based on runge–kutta physics-informed neural network. Nonlinear Dyn 2023;111(22):21117–30.
[21] Zhang S-L, Wang M-h, Zhao Y-C. Bright-dark rogue wave transition in coupled AB system via the physics-informed neural networks method. Commun Appl Math Comput Sci 2024;19(1):1–26.
[22] Pang G, Lu L, Karniadakis GE. fPINNs: fractional physics-informed neural networks. SIAM J Sci Comput 2019;41(4):A2603–A2626.
[23] Yang L, Meng X, Karniadakis GE. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. J Comput Phys 2021;425:109913.
[24] Wang S, Wang H, Perdikaris P. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. Sci Adv 2021;7(40):eabi8605.
[25] Li Z, Zheng H, Kovachki N, Jin D, Chen H, Liu B, et al. Physics-informed neural operator for learning partial differential equations. ACM / IMS J Data Sci 2024;1(3):9:1–9:27.
[26] Lan P, Su J-j, Ma X-y, Zhang S. Application of improved physics-informed neural networks for nonlinear consolidation problems with continuous drainage boundary conditions. Acta Geotech 2024;19(1):495–508.
[27] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nat Rev Phys 2021;3(6):422–40.
[28] Cuomo S, Di Cola VS, Giampaolo F, Rozza G, Raissi M, Piccialli F. Scientific machine learning through physics–informed neural networks: where we are and what's next. J Sci Comput 2022;92(3):88.
[29] Baydin AG, Pearlmutter BA, Radul AA, Siskind JM. Automatic differentiation in machine learning: a survey. J Mach Learn Res 2018;43(5):A3055–A3081.
[30] Wang S, Teng Y, Perdikaris P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM J Sci Comput 2021.
[31] Krishnapriyan A, Gholami A, Zhe S, Kirby R, Mahoney MW. Characterizing possible failure modes in physics-informed neural networks. In: Advances in neural information processing systems; vol. 34. Curran Associates, Inc.; 2021, pp. 26548–60.
[32] Cao W, Zhang W. An analysis and solution of ill-conditioning in physics-informed neural networks. J Comput Phys 2025;520:113494.
[33] Zhang Z, Xiong X, Zhang S, Wang W, Yang X, Zhang S, et al. A pseudo-time stepping and parameterized physics-informed neural network framework for Navier–Stokes equations. Phys Fluids 2025;37(3):033612.
[34] Kůrková V. Kolmogorov's theorem and multilayer neural networks. Neural Netw 1992;5(3):501–6.
[35] Sára Pusztaházi L, Eigner G, Csiszár O. Parametric activation functions for neural networks: a tutorial survey. IEEE Access 2024;12:168626–44.

[36] Jagtap AD, Kawaguchi K, Karniadakis GE. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. J Comput Phys 2020;404:109136.
[37] Liu Z, Wang Y, Vaidya S, Ruehle F, Halverson J, Soljacic M, et al. KAN: Kolmogorov–Arnold networks. In: The thirteenth international conference on learning representations. 2024,.
[38] Li Z. Kolmogorov–Arnold networks are radial basis function networks. 2024. arXiv:2405.06721.
[39] Meshir JD, Palafox A, Guerrero EA. On the study of frequency control and spectral bias in wavelet-based Aolmogorov Arnold networks: a path to physics-informed KANs. 2025. arXiv:2502.00280.
[40] Sidharth SS, Keerthana AR, Gokul R, Anas KP. Chebyshev polynomial-based Kolmogorov–Arnold networks: an efficient architecture for nonlinear function approximation. 2024. arXiv:2405.07200.
[41] Chen C, Hou K, Li J, Zhang W. Physics-informed Kolmogorov–Arnold network with chebyshev polynomials for fluid mechanics. Phys Fluids 2024;36(11):117116.
[42] Toscano JD, Oommen V, Varghese AJ, Zou Z, Ahmadi Daryakenari N, Wu C, et al. From pinns to pikans: recent advances in physics-informed machine learning. Mach Learn Comput Sci Eng 2025;1(1):1–43.
[43] Toscano JD, Yang Z, Karniadakis GE. Kkans: Kurkova–Kolmogorov–Arnold networks and their learning dynamics. Neural Netw 2024;182:106893.
[44] Li A, Kontogiannis A, Karniadakis GE, Maxey MR. Aivt: inference of turbulent thermal convection from measured 3D velocity data by physics-informed Kolmogorov–Arnold networks. Sci Adv 2024;10(50):eads5236.
[45] Chihara TS. An Introduction to Orthogonal Polynomials. 2011.
[46] Littlejohn LL, Krall AM. Orthogonal polynomials and higher order singular Sturm–Liouville systems. Acta Appl Math 1989;17(2):99–170.
[47] Shukla K, Toscano JD, Wang Z, Zou Z, Karniadakis GE. A comprehensive and FAIR comparison between MLP and KAN representations for differential equations and operator networks. Comput Methods Appl Mech Eng 2024;431:117290.
[48] Koenig BC, Kim S, Deng S. KAN-ODEs: Kolmogorov–Arnold network ordinary differential equations for learning dynamical systems and hidden physics. Comput Methods Appl Mech Eng 2024;432:117397.
[49] Wang Y, Sun J, Bai J, Anitescu C, Eshaghi MS, Zhuang X, et al. Kolmogorov–Arnold-informed neural network: a physics-informed deep learning framework for solving forward and inverse problems based on Kolmogorov–Arnold networks. Comput Methods Appl Mech Eng 2025;433:117518.
[50] Raissi M, Yazdani A, Karniadakis GE. Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. Science 2020;367(6481):1026–30.
[51] Liu L, Liu S, Xie H, Xiong F, Yu T, Xiao M, et al. Discontinuity computing using physics-informed neural networks. J Sci Comput 2023;98(1):22.
[52] Kiyani E, Shukla K, Urbán JF, Darbon J, Karniadakis GE. Which optimizer works best for physics-informed neural networks and Kolmogorov–Arnold networks? 2025. arXiv:2501.16371.
[53] Urbán JF, Cuomo S, Karniadakis GE. Unveiling the optimization process of physics informed neural networks: how accurate and competitive can PINNs be? J Comput Phys 2024;520:113045.
[54] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014. arXiv:1412.6980.
[55] Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. Math Program 1989;45(1):503–28.
[56] Wang S, Li B, Chen Y, Perdikaris P. Piratenets: physics-informed deep learning with residual adaptive networks. J Mach Learn Res 2024;25:1–51.
[57] Guo B-Y, Shen J, Wang L-L. Generalized Jacobi polynomials/functions and their applications. Appl Numer Math 2009;59(5):1011–28.
[58] Botella O, Peyret R. Benchmark spectral results on the lid-driven cavity flow. Comput Fluids 1998;27(4):421–33.
[59] Boyd JP. Chebyshev and Fourier spectral methods. Courier Corporation; 2001.
[60] Solsvik J, Jakobsen HA. Effects of Jacobi polynomials on the numerical solution of the pellet equation using the orthogonal collocation, galerkin, tau and least squares methods. Comput Chem Eng 2012;39:1–21.
[61] Chen Y, Ismail M. Jacobi polynomials from compatibility conditions. Proc Am Math Soc 2005;133(2):465–72.