
Deep vs. Shallow: Benchmarking Physics-Informed Neural Architectures on the Biharmonic Equation

Akshay Govind Srinivasan

Department of Mechanical Engineering
Indian Institute of Technology, Madras
me22b102@smail.iitm.ac.in

Vikas Dwivedi

Creatis Biomedical Imaging Laboratory
INSA-Lyon, France
vikas.dwivedi@creatis.insa-lyon.fr

Balaji Srinivasan

Department of Data Science and AI
Indian Institute of Technology, Madras
sbalaji@iitm.ac.in

Abstract

Partial differential equation (PDE) solvers are fundamental to engineering simulation. Classical mesh-based approaches (finite difference/volume/element) are fast and accurate on high-quality meshes but struggle with higher-order operators and complex, hard-to-mesh geometries. Recently developed physics-informed neural networks (PINNs) and their variants are mesh-free and flexible, yet compute-intensive and often less accurate. This paper systematically benchmarks RBF-PIELM, a rapid PINN variant—an extreme learning machine with radial-basis activations—for higher-order PDEs. RBF-PIELM replaces PINNs’ time-consuming gradient descent with a single-shot least-squares solve. We test RBF-PIELM on the fourth-order biharmonic equation using two benchmarks: lid-driven cavity flow (streamfunction formulation) and a manufactured oscillatory solution. Our results show up to $350\times$ faster training than PINNs and over $10\times$ fewer parameters for comparable solution accuracy. Despite surpassing PINNs, RBF-PIELM still lags mature mesh-based solvers and its accuracy degrades on highly oscillatory solutions, highlighting remaining challenges for practical deployment. We open-source our code for reproducibility and future extensions¹

1 Introduction

Numerical solution of ODEs and PDEs underpins modeling in fluid and solid mechanics. Classical discretization methods—finite difference (FDM) Strikwerda [2004], finite element (FEM) Hughes [2012], and finite volume (FVM) Versteeg and Malalasekera [2007]—offer high accuracy but incur significant computational cost due to their larger stencils, and their reliance on expensive mesh generation in complex or large-scale domains Yagawa [2011]. This motivates a need for developing mesh-free methods for solving ODEs and PDEs for complex domain. Physics-Informed Neural Networks (PINNs) Raissi et al. [2019] provide a mesh-free alternative by embedding governing equations into neural network training. While they have demonstrated success across a range of applications, their training times are often significantly higher than those of state-of-the-art numerical solvers for achieving comparable accuracy McGreivy and Hakim [2024]. Hence making them inefficient as compared to traditional solvers. Moreover, challenges remain in terms of sensitivity to hyperparameter choices and the lack of interpretability in their learned representations Krishnapriyan et al. [2021].

¹The link to the Anonymous Repository can be found here

Dwivedi and Srinivasan introduced Physics-Informed Extreme Learning Machines (PIELMs) as an efficient alternative to PINNs, which merge the physics-based loss of PINNs with the shallow architecture of Extreme Learning Machines (ELMs) Huang et al. [2006a]. Unlike gradient-based training Shalev-Shwartz et al. [2017], ELMs fix input weights randomly and compute output weights via a pseudo-inverse, yielding orders-of-magnitude faster training while retaining universal approximation guarantees Huang et al. [2006a,b]. PIELMs thus provide rapid and data-efficient PDE solvers Dwivedi and Srinivasan [2020], Xu et al. [2022]. Since PIELMs initialize their input weights randomly, the resulting hidden features lack physical interpretability and exhibit limited alignment with the underlying physics, hence cannot be initialized in a physics-informed manner.

Dwivedi et al. introduced Radial Basis Function-based PIELMs (RBF-PIELMs), which extend PIELMs by replacing random hidden features with localized Radial Basis Functions (RBFs) Buhmann [2003], Schaback [2006]. RBFs offer interpretable, physics-aware activations: each hidden unit corresponds to a localized “receptive field,” and centers can be aligned with domain geometry, boundary layers, or measurement data Kansa [1990]. Their widths can be tuned to resolve sharp gradients such as shear layers or shocks. Moreover, RBF networks retain universal approximation guarantees Park and Sandberg [1991], preserving theoretical expressivity while maintaining the efficiency of PIELMs through a closed-form solution.

Contributions This paper investigate the applicability of RBF-PIELMs as a fast, interpretable, and flexible solver for linear PDEs. Specifically:

- We investigate the applicability of RBF-PIELMs with geometry-aware initialization for solving higher order PDE.
- We benchmark against PINNs on the lid-driven cavity problem, showing similar accuracy with much lower training cost and model complexity.
- We demonstrate expressivity of RBF-PIELM by solving a complex biharmonic problem Pan et al. [2025] and evaluate it using the Method of Manufactured Solutions (MMS).

The remainder of this paper is organized as follows: Section 2 presents the mathematical formulation of RBF-PIELM; Section 3.1 presents the lid-driven cavity benchmark presented in Marchi et al.; Section 3.2 introduces the MMS study presented in Pan et al.; Section 4 reports and discusses results and concludes with future directions.

2 Methodology: RBF-PIELM

Let $u : \Omega \subset \mathbb{R}^m \rightarrow \mathbb{R}^n$ be a function that satisfies Equation 1 where $\mathcal{L}(u)(x)$ is a linear differential operator in a given domain Ω

$$\mathcal{L}(u)(x) + f(x) = 0, \quad x \in \Omega, \quad (1)$$

with boundary operator \mathcal{B} and data g on $\partial\Omega$. In RBF-PIELM, the function approximation is:

$$\hat{u}(x) = \sum_{i=1}^{N^*} c_i \phi_i(x), \quad \phi_i(x) = \exp\left(-\frac{\|x-x_i\|^2}{2\sigma_i^2}\right), \quad (2)$$

where x_i and σ_i denote the center and width of the i -th RBF Buhmann [2003]. Centers and widths may be chosen randomly or in a problem-aware like choosing more RBFs near walls and boundary layers, or they can be initialized in a data-driven manner Dwivedi et al. [2025b]. Given that the interior collocation points are represented as $\{x_j^\Omega\}$ and boundary points as $\{x_k^\partial\}$, the residuals are

$$\mathcal{R}_\Omega(x) = \mathcal{L}(\hat{u})(x) + f(x), \quad x \in \Omega, \quad (3)$$

$$\mathcal{R}_\partial(x) = \mathcal{B}(\hat{u})(x) - g(x), \quad x \in \partial\Omega. \quad (4)$$

Enforcing residuals at collocation points yields an over-constrained linear system (assuming that $N^* < |\Omega| + |\partial\Omega|$) as in Equation 5 with entries of A defined by the PDE and boundary operators.

$$A c = b, \quad (5)$$

The coefficients c are then obtained via the penrose pseudo-inverse. These coefficients can then be substituted in Equation 2 to obtain approximate solution to Equation 1.

3 Numerical Experiments

3.1 Biharmonic Equation with Smooth Solution: Lid-Driven Cavity Flow

The lid-driven cavity problem is a classical benchmark in incompressible flow simulations, and in this work we adopt its stream function–vorticity (biharmonic) formulation. The biharmonic equation arises in modeling mixing in cavities, micromechanical flows, and in solid mechanics through the Airy stress function Radice [2021]. Its fourth-order nature makes it challenging to solve using mesh-free methods due to the need for accurately enforcing multiple boundary conditions and handling higher derivatives. We use it both for its ubiquity and difficulty, benchmarking our method by comparing centerline velocities with the results of Marchi et al.. The velocity $\mathbf{u} = (u, v)^T$ satisfy

$$-\nabla^2 \mathbf{u} + \nabla p = 0, \quad \nabla \cdot \mathbf{u} = 0. \quad (6)$$

Introducing the stream function $\psi(x, y)$ via $u = \frac{\partial \psi}{\partial y}$, $v = -\frac{\partial \psi}{\partial x}$, reduces the equations to the biharmonic form as in Equation 7.

$$\psi_{xxxx} + 2\psi_{xxyy} + \psi_{yyyy} = 0, \quad (x, y) \in [0, 1]^2, \quad (7)$$

with no-slip boundary conditions. The lid motion is enforced by $\psi_y(x, 1) = 1$, while all other walls remain fixed. The definitions of Boundary Conditions is given in Appendix Section A.

Results and Discussion

- **Physics-Aware Initialization** To accurately capture the sharp velocity gradients near the cavity walls, the collocation points are distributed with Chebyshev spacing (Refer Appendix Section B) and the RBF Kernels are over-sampled near boundaries, ensuring higher resolution near boundaries. The RBF standard deviation is chosen using a heuristic: $\sigma = 0.3 + 0.93 \left(\frac{L_{\min}}{L_{\max}} \right)$, where L_{\min} is the distance of the kernel center to the nearest wall and L_{\max} the maximum possible distance in the domain. Further details about tuning the parameters of the heuristics is discussed in Appendix Section F. This physically motivated choice ensures narrower kernels near boundaries and broader ones in the interior, providing more physics-aware initialization of input layer parameters.
- **PIELM vs PINN** Figure 1a and 1b compares the centerline velocities obtained from PINN, RBF-PIELM and Marchi et al.. We observe that RBF-PIELM outperforms PINN and closely matches the reference solution. As shown in Table 1, RBF-PIELM computes the solution **350× faster** in 0.4347s compared to 151.77s for PINNs, using **13.2× fewer parameters** and **1.8× fewer collocation points**, while attaining lower residual errors.
- **Effect of Physics Aware Initialization** Figure 1a and 1b compares the centerline velocities obtained from RBF-PIELM with and without Physics Aware Initialization (PAI). We observe that RBF-PIELM with PAI outperforms RBF-PIELM without PAI and closely matches the reference solution. As shown in Table 1, RBF-PIELM with PAI has **17.7% lower residual** as compared to RBF-PIELM without PAI.

Method	Time (s) ↓	Parameters ↓	Collocation Pts ↓	Residual ↓
PINN	151.77	9921	4922	9.84×10^{-3}
RBF-PIELM w/o PAI	0.4260	750	2688	6.68×10^{-3}
RBF-PIELM with PAI	0.4347	750	2688	5.44×10^{-3}

Table 1: Performance comparison for Lid-Driven Cavity. RBF-PIELM attains higher efficiency with fewer parameters and lower residual error.

3.2 Biharmonic Equation with Oscillatory Solution

To further assess the capability of RBF-PIELM, we consider the Method of Manufactured Solutions (MMS), which enables direct accuracy testing by prescribing an exact solution and deriving the corresponding biharmonic problem. In particular, we benchmark against Example 5.2 from Pan et al.

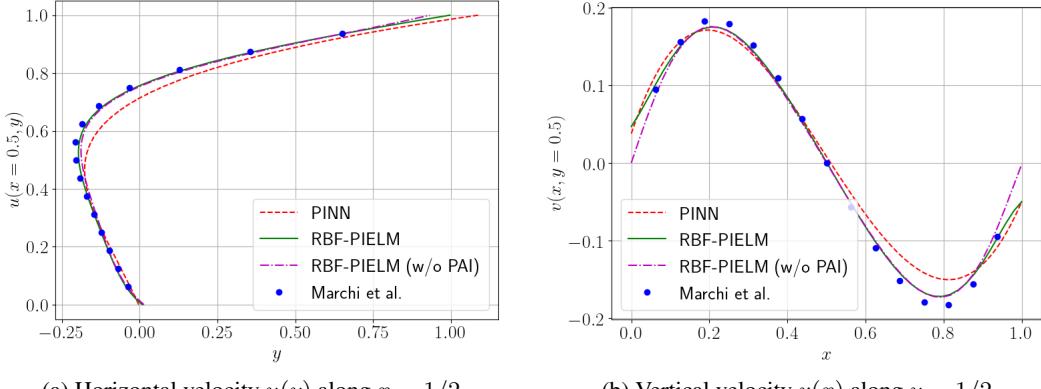


Figure 1: Centerline Velocity Profiles: RBF-PIELM vs. PINNs. RBF-PIELM attains better accuracy while training $350\times$ faster than PINNs.

[2025], where a fourth-order compact finite-difference scheme was evaluated. The manufactured solution involves oscillatory components (with $k_1 = k_2 = 10$), making it a challenging test for approximation methods. We provide the derivation and exact boundary conditions imposed for the problem in Appendix Section C. We sample 60×60 collocation points in the domain to capture the complex oscillatory nature of the problem. Additionally, we also increase the number of RBFs to 2000 to represent the complex solution. Unlike the lid-driven cavity case, which highlights physical relevance and efficiency, this experiment probes whether RBF-PIELM can approximate complex functions. Figure 2 compares the exact solution, the RBF-PIELM prediction, and the error distribution. Despite the solution’s complexity, RBF-PIELM, achieves a mean error of 3.46×10^{-2} , thereby demonstrating its expressivity. RBF-PIELM solves the problem in 6.4 seconds with above mentioned configuration.

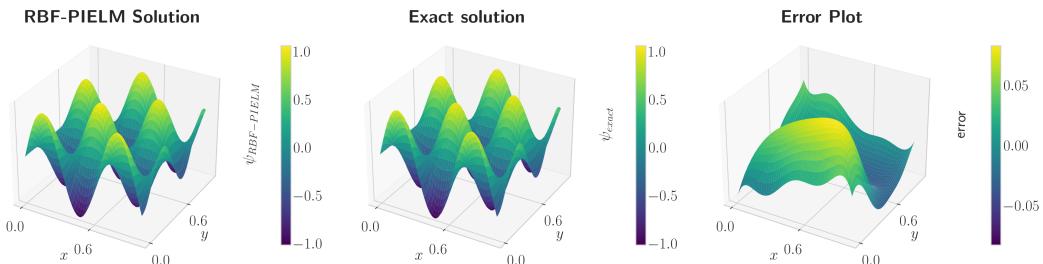


Figure 2: Manufactured biharmonic solution ($k_1 = k_2 = 10$): (a) RBF-PIELM solution, (b) Exact solution and (c) Distribution of absolute error. RBF-PIELM closely matches the oscillatory exact solution with small errors.

4 Conclusion

This work assessed RBF-PIELM as a lightweight, interpretable alternative to gradient-based PINNs for fourth-order PDEs. We evaluated two benchmarks: lid-driven cavity flow in a streamfunction formulation and a manufactured oscillatory solution of the biharmonic equation. On the cavity problem, RBF-PIELM achieved a $\sim 350\times$ training speedup over PINNs (0.4347 s vs. 151.77 s), used $13.2\times$ fewer parameters, and delivered 44.7% lower error. We also showed that PAI produced better results. On the manufactured case, it reproduced the analytical field competitively, though accuracy degraded as the oscillatory content increased. These gains stem from physics-informed initialization of RBF centers (enhancing interpretability) and a single-shot least-squares solve for output weights (eliminating costly iterative training). Despite clear advantages over PINNs, RBF-PIELM still trails mature mesh-based solvers in wall-clock performance and robustness on highly oscillatory fields (Refer Appendix Section D), underscoring opportunities for hybrid (FEM-PIELM) formulations and residual adaptive basis refinement.

References

- John C Strikwerda. *Finite difference schemes and partial differential equations*. SIAM, 2004.
- Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.
- Henk Kaarle Versteeg and Weeratunge Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson education, 2007.
- Genki Yagawa. Free mesh method: fundamental conception, algorithms and accuracy study. *Proc. Jpn. Acad. Ser. B Phys. Biol. Sci.*, 87(4):115–134, 2011.
- Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- Nick McGreivy and Ammar Hakim. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. *Nature Machine Intelligence*, 6(10):1256–1269, September 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00897-5. URL <http://dx.doi.org/10.1038/s42256-024-00897-5>.
- Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34:26548–26560, 2021.
- Vikas Dwivedi and Balaji Srinivasan. Physics informed extreme learning machine (pielm)—a rapid method for the numerical solution of partial differential equations. *arXiv preprint arXiv:1907.03507*, 2019.
- Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006a.
- Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. *arXiv preprint arXiv:1703.07950*, 2017. URL <https://arxiv.org/abs/1703.07950>.
- Guang-Bin Huang, Lei Chen, and Chai Quek Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 17(4):879–892, 2006b. doi: 10.1109/TNN.2006.875977.
- Vikas Dwivedi and Balaji Srinivasan. Physics informed extreme learning machine (pielm)—a rapid method for the numerical solution of partial differential equations. *Neurocomputing*, 391:96–118, 2020.
- Liu Xu et al. Bayesian physics-informed extreme learning machine for forward and inverse pde problems with noisy data. *arXiv preprint arXiv:2205.06948*, 2022.
- Vikas Dwivedi, Bruno Sixou, and Monica Sigovan. Curriculum learning-driven pielms for fluid flow simulations. *Neurocomputing*, 650:130924, 2025a. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2025.130924>. URL <https://www.sciencedirect.com/science/article/pii/S0925231225015966>.
- Martin D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, Cambridge, 2003. ISBN 9780521101332.
- Robert Schaback. Kernel techniques: from machine learning to meshless methods. *Acta Numerica*, 15:543–639, 2006.
- EJ Kansa. Multiquadratics—a scattered data approximation scheme with applications to computational fluid-dynamics—i surface approximations and partial derivative estimates. *Computers & Mathematics with Applications*, 19(8-9):127–145, 1990.
- J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2):246–257, June 1991. ISSN 1530-888X. doi: 10.1162/neco.1991.3.2.246. URL <http://dx.doi.org/10.1162/neco.1991.3.2.246>.

Kejia Pan, Jin Li, Zhilin Li, and Hongling Hu. A fourth order mixed compact finite difference scheme for biharmonic equations. *Journal of Computational Physics*, 539:114254, 2025. ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2025.114254>. URL <https://www.sciencedirect.com/science/article/pii/S0021999125005376>.

Carlos Marchi, Roberta Suero, and Luciano Araki. The lid-driven square cavity flow: Numerical solution with a 1024 x 1024 grid. *Journal of The Brazilian Society of Mechanical Sciences and Engineering - J BRAZ SOC MECH SCI ENG*, 31, 07 2009. doi: 10.1590/S1678-58782009000300004.

Vikas Dwivedi, Balaji Srinivasan, Monica Sigovan, and Bruno Sixou. Kernel-adaptive pi-elms for forward and inverse problems in pdes with sharp gradients, 2025b. URL <https://arxiv.org/abs/2507.10241>.

J. Radice, J. A biharmonic polynomial airy stress function for the square-end adhesive layer and sandwich core. *Mechanics Research Communications*, 113:103684, 2021.

A Boundary Conditions for Lid-Driven Cavity

The boundary condition imposed are as depicted in Figure 3.

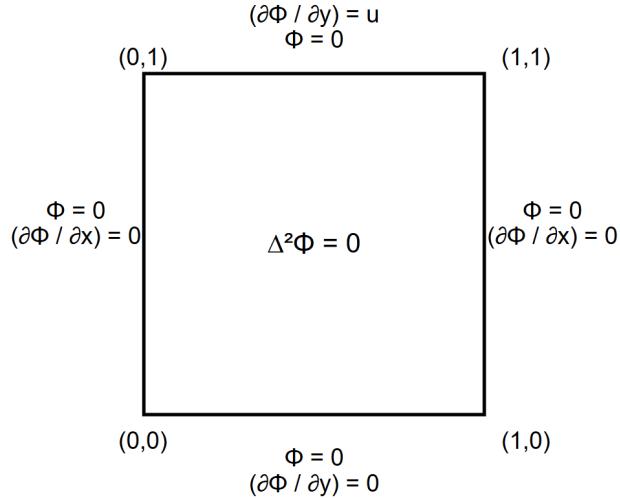


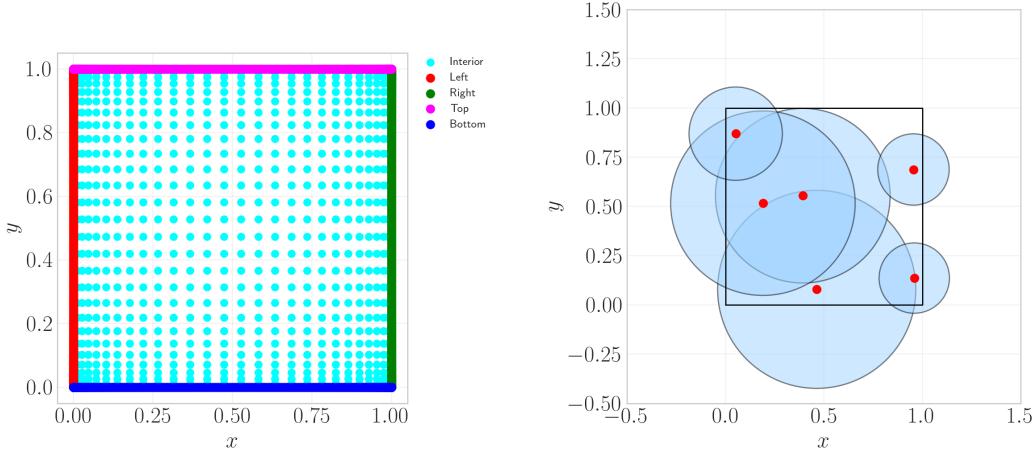
Figure 3: Lid-driven cavity benchmark: moving top wall ($u = 1$) and stationary side/bottom walls.

B Chebyshev Spacing for Sampling and Kernel Placement

In our experiments, collocation points are distributed using Chebyshev spacing along each coordinate axis. For N points in the interval $[0, 1]$, the Chebyshev nodes are defined as

$$x_j = \frac{1}{2} \left[1 - \cos \left(\frac{j\pi}{N-1} \right) \right], \quad j = 0, 1, \dots, N-1. \quad (8)$$

This distribution concentrates points near domain boundaries while spacing them more sparsely in the interior. Such clustering is advantageous in the lid-driven cavity problem, where strong velocity gradients occur near walls and uniform sampling may under-resolve them. Chebyshev nodes are also well established in spectral and collocation methods for reducing interpolation error and improving stability. For mesh-free solvers like RBF-PIELM, this spacing provides boundary refinement without ad hoc clustering, balancing interior coverage with boundary resolution and leading to more efficient, accurate solutions (Figure 4a).



(a) Distribution of collocation and boundary points. (b) Visualization of RBF kernels. RBF centers (red) with widths encoded by the radii of blue disks.

Figure 4: Collocation/boundary points and RBF width visualization for Test Case 1.

C Manufactured Solution Details for Example 5.2 from Pan et al.

In Section 3.2, the method of manufactured solutions is employed from Pan et al. to rigorously validate the accuracy of our numerical solver for biharmonic equations. Specifically, we select an exact, oscillatory solution of the form

$$u(x, y) = \sin(k_1 x) \cos(k_2 y), \quad (x, y) \in (0, 1)^2, \quad (9)$$

where k_1 and k_2 are prescribed parameters (here, both set to 10).

To construct the corresponding source term $f(x, y)$ required for the biharmonic equation

$$\Delta^2 u(x, y) = f(x, y), \quad (10)$$

we substitute the analytic solution $u(x, y)$ into the biharmonic operator. In two dimensions, the biharmonic operator is given by

$$\Delta^2 u = \frac{\partial^4 u}{\partial x^4} + 2 \frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4}. \quad (11)$$

Applying this operator to $u(x, y)$,

$$\frac{\partial^2 u}{\partial x^2} = -k_1^2 \sin(k_1 x) \cos(k_2 y), \quad (12)$$

$$\frac{\partial^2 u}{\partial y^2} = -k_2^2 \sin(k_1 x) \cos(k_2 y), \quad (13)$$

and continuing to fourth derivatives,

$$\frac{\partial^4 u}{\partial x^4} = k_1^4 \sin(k_1 x) \cos(k_2 y), \quad (14)$$

$$\frac{\partial^4 u}{\partial y^4} = k_2^4 \sin(k_1 x) \cos(k_2 y), \quad (15)$$

$$\frac{\partial^4 u}{\partial x^2 \partial y^2} = k_1^2 k_2^2 \sin(k_1 x) \cos(k_2 y), \quad (16)$$

which yields

$$f(x, y) = \Delta^2 u(x, y) \quad (17)$$

$$= [k_1^4 + 2k_1^2 k_2^2 + k_2^4] \sin(k_1 x) \cos(k_2 y). \quad (18)$$

This manufactured source term $f(x, y)$ is used in the numerical experiments to assess the solver's accuracy, with the exact solution available for direct comparison.

Boundary Conditions The boundary conditions imposed are Dirichelet in nature and are obatined from function values.

D Limitations of RBF-PIELM

We here show a more complex solution from Pan et al. [2025] by setting $k_1 = k_2 = 20$. We present the result in Figure 5. We observe that RBF-PIELM faces several challenges in representing the solution more granularly.

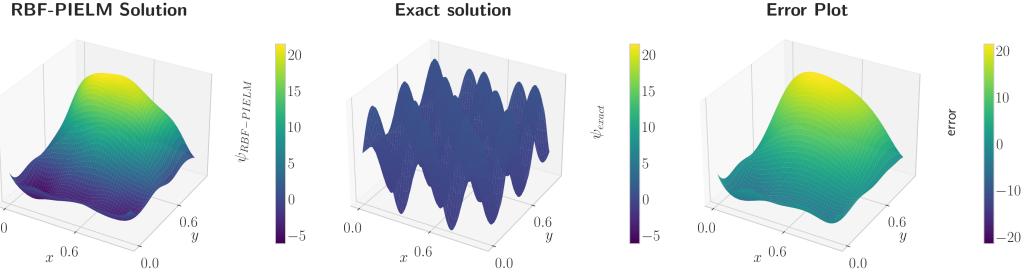


Figure 5: Manufactured biharmonic solution ($k_1 = k_2 = 20$): (a) RBF-PIELM solution, (b) Exact solution, and (c) Distribution of absolute error. RBF-PIELM breaks down on this highly oscillatory case.

E Additional Results for Test Case 1

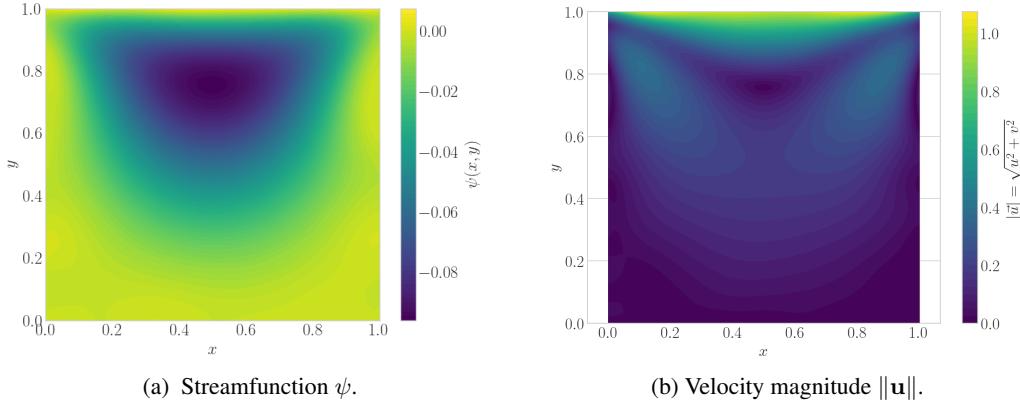
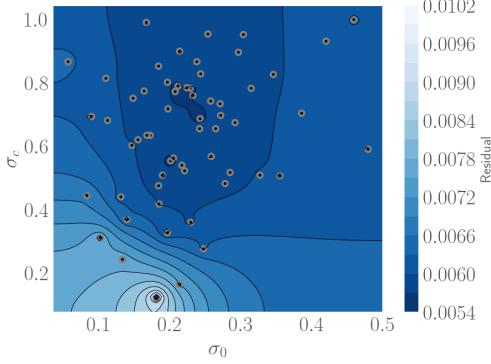


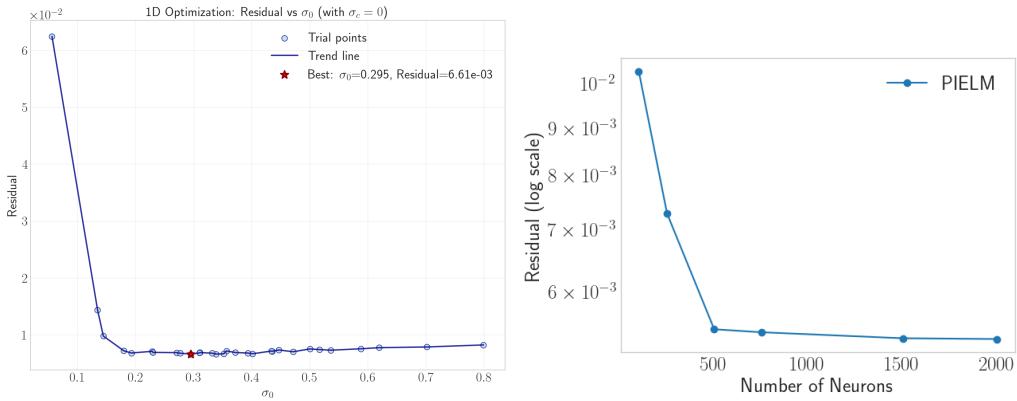
Figure 6: RBF-PIELM predictions of streamfunction ψ and velocity magnitude field $\|u\|$.

F RBF Width Tuning

To get the best performance from RBF-PIELM we tune the hyperparameters of σ function, namely, σ_0 and σ_c . Figure 7a shows the contours of residuals with x and y axis being different axes of σ_0 and σ_c respectively. We see that tuning the hyperparameters decreases the residual by around 50%. We also show the variations of Residual with number of Neurons/RBFs in Figure 7c. We see that as the number of neurons increases, at a certain point the residual saturates.



(a) Residual contours over (σ_0, σ_c) .



(b) Residual as a function of σ_0 (with σ_c fixed). (c) Residual as a function of Number of Neurons (N^*).

Figure 7: Hyperparameter sensitivity of the residual: tuning σ_0 and σ_c reduces the residual by approximately 50%.

G Neural Network Specifications

For training PINNs, we use the PyTorch² library with Nvidia Tesla P100, offered through Kaggle. Further the network used has 5 layers with the following number of neurons: [2, 40, 120, 40, 1]. The optimizer used was Adam with learning rate set to 5×10^{-3} .

H Compute Specifications

For RBF-PIELM Experiments, we run the experiment only on a CPU namely, a Ryzen 7 5800H with 16GB RAM with a single process. For PINN based experiments, we use Nvidia Tesla P100, offered through Kaggle.

²<https://pytorch.org/>