Curriculum-Learning PIELMs for Hemodynamic Flows

Vikas Dwivedi VIKAS.DWIVEDI@CREATIS.INSA-LYON.FR
Monica Sigovan MONICA.SIGOVAN@CREATIS.INSA-LYON.FR
Bruno Sixou BRUNO.SIXOU@INSA-LYON.FR
Creatis Biomedical Imaging Laboratory, INSA-Lyon, France

Abstract

Physics-informed machine learning (PIML) has emerged as a promising approach for solving partial differential equations (PDEs), but its adoption in real-world applications is often constrained by trade-offs between speed, accuracy, and interpretability. In this work, we evaluate a curriculum learning–driven physics-informed extreme learning machine (CL-PIELM) and benchmark it against FEniCS, a state-of-the-art finite element method (FEM) solver for viscous two-dimensional fluid flow in a hemodynamically relevant bifurcation geometry, where elevated flow speeds lead to sharp pressure jumps. The main feature of CL-PIELM is to reformulate a nonlinear optimization problem as a sequence of increasingly complex linear optimization problems. Our findings show that CL-PIELM offers interpretable parameter selection and extends traditional PIELM to iteratively solve nonlinear PDEs, achieving reasonable accuracy within moderate Reynolds number regimes. While FEM remains faster and more accurate in the tested configurations, the study demonstrates the potential of CL-PIELM as a flexible, physics-informed framework that can be further enhanced for large-scale and high Reynolds number flow scenarios.

Keywords: Physics-Informed Neural Network, Physics-Informed Extreme Learning Machine, Radial Basis Functions, Navier-Stokes Equation

1. Introduction

Partial differential equations (PDEs) describe fundamental physical laws governing spatio-temporal dynamics. Classical numerical solvers such as finite element (FEM), finite difference (FDM), and finite volume (FVM) methods provide accurate solutions, but often at the cost of high mesh generation and computational effort in complex geometries. Recently, physics-informed neural networks (PINNs) [5, 7] have been proposed as mesh-free alternatives, embedding governing equations directly into the training loss. While effective, PINNs are hindered by slow training and limited interpretability in network design [6, 8].

Physics-Informed Extreme Learning Machines (PIELMs) [2] offer a faster alternative by exploiting the linearity of single-hidden-layer networks in their output weights, enabling closed-form training without backpropagation. This yields significant speedups, especially for linear PDEs, but two major challenges remain: extending PIELMs to nonlinear PDEs, and developing systematic, physically interpretable initialization schemes. Prior remedies, such as nonlinear least-squares solvers or evolutionary optimization [1], are computationally heavy and offer little interpretability.

Curriculum Learning–driven PIELMs (CL-PIELMs) [4] address these limitations by combining physics-informed initialization of radial basis functions with a curriculum strategy that **solves a nonlinear optimization problem (Navier-Stokes equations) via a sequence of its linearized**

versions at progressively higher Reynolds numbers. Initial studies have demonstrated promising accuracy and speed on benchmark flows.

In this work, we evaluate CL-PIELM in a hemodynamically relevant bifurcating geometry and benchmark its performance against FEM, a mature numerical solver. The study assesses accuracy, efficiency, and failure modes across Reynolds numbers, highlighting both the potential and current limitations of CL-PIELMs as a physics-informed framework for nonlinear fluid dynamics.

2. Methodology

Governing Equations. We consider the steady, incompressible Navier–Stokes equations in 2D:

$$\nabla \cdot \mathbf{V} = 0, \quad (\mathbf{V} \cdot \nabla)\mathbf{V} = -\nabla p + \tfrac{1}{Re}\nabla^2 \mathbf{V},$$

with velocity $\mathbf{V} = (u, v)^T$, pressure p, and Reynolds number $Re = UL/\nu$.

PIELM Ansatz. Unlike PINNs that use deep networks, PIELMs employ a single hidden-layer feedforward network with fixed input-layer parameters, yielding linearity in output weights. For (u, v, p) we adopt an RBF-based ansatz

$$\hat{q}(x, y) = \sum_{k=1}^{N^*} \phi(z_k(x, y))\, c_{q,k}, \quad q \in \{u, v, p\},$$

where $\phi(z) = e^{-z^2}$ and $z_k(x, y)$ is given by

$$z_k(x, y) = \sqrt{(m_k x + \alpha_k)^2 + (n_k y + \beta_k)^2}.$$

Input parameters for (u, v, p) are shared to respect pressure–velocity coupling.

Physics-Based Initialization. CL-PIELM assigns Gaussian-kernel–inspired parameters: kernel centers are denser near corners/walls, and widths are smaller in regions of sharp gradients. Once centers (α^*, β^*) and widths (σ_x, σ_y) are set, the input parameters are given by, $\alpha = -\alpha^*/(\sqrt{2}\,\sigma_x),\ \beta = -\beta^*/(\sqrt{2}\,\sigma_y),\ m = 1/(\sqrt{2}\,\sigma_x),\ n = 1/(\sqrt{2}\,\sigma_y)$. This replaces random initialization with a physically interpretable scheme. Dwivedi et al. [3] present an interesting application of physics-bound initialization in the context of medical image segmentation.

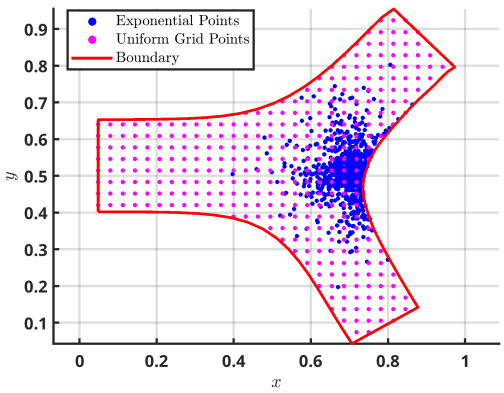
Curriculum Learning. To handle the nonlinear advection term, the Navier–Stokes system is solved incrementally in Re, approximating

$$(\mathbf{V} \cdot \nabla)\mathbf{V}\big|_{Re} \approx (\mathbf{V}_{Re-\delta} \cdot \nabla)\mathbf{V}_{Re}.$$

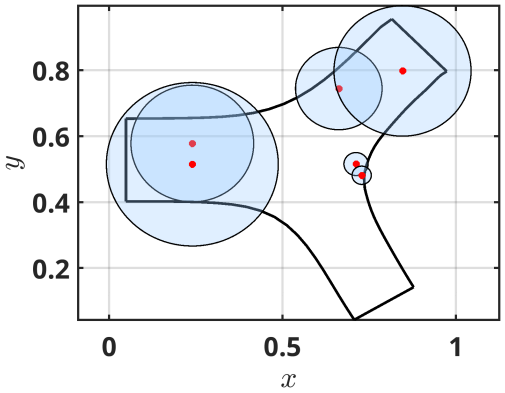
This transforms the nonlinear system into a sequence of linear least-squares problems.

Residual Equations. PDE and boundary residuals are evaluated at collocation points and assembled into an overdetermined block-linear system $\boldsymbol{Mc} = 0$, solved via pseudo-inverse. The detailed expressions are given in Appendix-A.

Optimization. Starting from $Re = 0$, each curriculum step computes the flow in closed form (single least-squares solve) and propagates it to the next level, $Re \leftarrow Re + \delta$, as spatially varying coefficients in the linear advection terms. This eliminates iterative backpropagation and thereby yields substantial speedups over PINNs.



(*a*) Sampling-point distribution.



(*b*) Spatial variation of RBF kernel widths.

Figure 1: (a) Sampling-point distribution and (b) RBF kernel width variation.

3. Problem Setup

The bifurcation geometry (Fig. 1(*a*)subfigure) has an inlet radius $R_{\text{in}} = 0.1254$ and two outlets, oriented at 105°, with radii $R_{\text{out},1} = 0.2240$ and $R_{\text{out},2} = 0.1978$. Boundary conditions are: inlet with parabolic velocity profile $\mathbf{V} = \left(U_{\max}[1 - (\frac{y-y_c}{R_{\text{in}}})^2],\, 0\right)^T$, where $U_{\max} = 0.1$ and $y_c = 0.5275$; outlets with $p = 0$ and $\partial u/\partial n = \partial v/\partial n = 0$; and walls with no-slip $\mathbf{V} = (0, 0)^T$. The Reynolds number is $Re = \frac{U_{\max} R_{\text{in}}}{\nu}$. This configuration becomes challenging at higher Re due to the sharp pressure gradient that develops at the junction.

4. Results and Discussion

Figures 1(*a*)subfigure and 1(*b*)subfigure show collocation point placement and RBF widths, with finer resolution near the junction. The network used 1083 collocation and 124 boundary points, with curriculum steps of $\delta_{Re} = 0.1$. The exact distribution of RBF-width is given in Appendix-B. All the experiments are conducted in Matlab R2022b environment running in a 12th Gen Intel(R) Core(TM) i7-12700H, 2.30 GHz CPU and 16GB RAM Asus laptop.

Accuracy. Against FEM (12.7k nodes, 24.8k elements), CL-PIELM reproduced velocity and pressure fields with good agreement up to $Re = 50$. At $Re = 10$ and $Re = 50$, RMS errors in velocity magnitude and pressure were $\mathcal{O}(10^{-4})$, sufficient to capture essential flow features. Figure 2 compares CL-PIELM with FEM.

Limitations. At higher Re (e.g., 100) and especially sharp corners, the method struggled to resolve sharp pressure jumps at the bifurcation (Fig. 3). Runtime was $\sim$20 minutes for $Re = 50$, slower than FEM (seconds). Although one could repeat the study with alternative collocation layouts, the principle remains: if the collocation set and RBF basis are not adapted to the Reynolds number, the method will fail for physical reasons (e.g., insufficient resolution at higher Re) Improvements such as adaptive kernel placement or scalable solvers could enhance performance.

Overall, CL-PIELM demonstrates interpretable initialization, reasonable accuracy in moderate Re regimes, and a clear physics-based pathway for extending PIELMs to nonlinear PDEs.

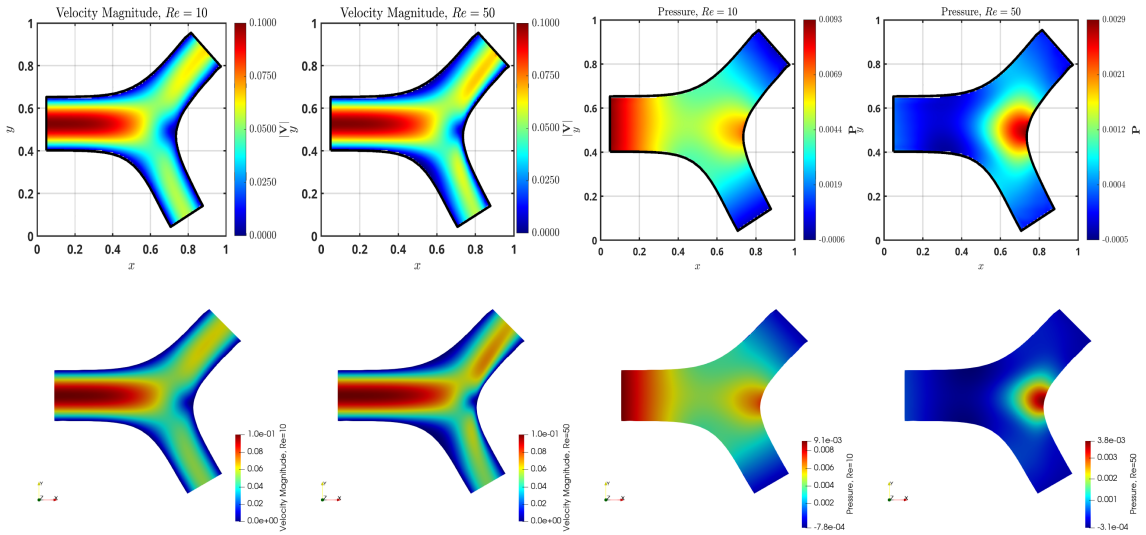


Figure 2: PIELM-predicted (top-row) and FEM (bottom-row) solutions for 2D bifurcation fluid flow at Re = 10, 50.

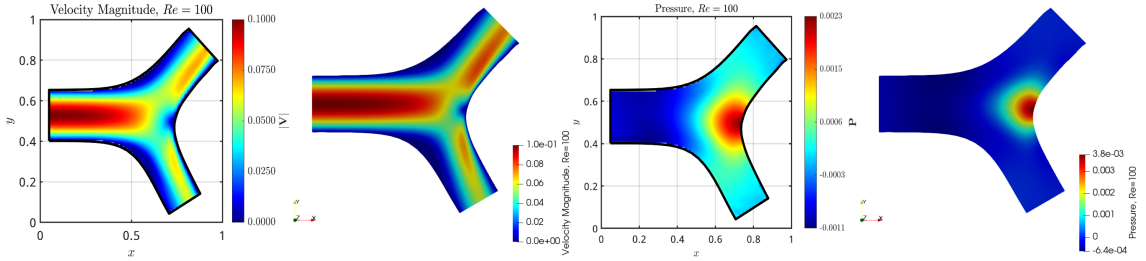


Figure 3: Limitations: PIELM does not capture the pressure jump at $Re = 100$.

5. Conclusion

This study evaluated CL-PIELMs for hemodynamic flow simulation, demonstrating their ability to solve nonlinear PDEs and validating performance on a bifurcation geometry. The method achieves reasonable accuracy at low Reynolds numbers but is slower than FEM, and struggles to capture sharp pressure gradients at higher Reynolds numbers. CL-PIELMs show promise as a physics-informed framework for complex flows, but unlocking their full potential will require targeted refinements to match the speed and accuracy of mature numerical solvers. Future work will focus on adapting the RBF configuration and sampling strategy to improve accuracy and efficiency.

References

[1] Suchuan Dong and Jielin Yang. On computing the hyperparameter of extreme learning machines: Algorithm and application to computational pdes, and comparison with classical and

high-order finite elements. *Journal of Computational Physics*, 463:111290, 2022. ISSN 0021-9991. URL https://doi.org/10.1016/j.jcp.2022.111290.

[2] 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. ISSN 0925-2312. URL https://doi.org/10.1016/j.neucom.2019.12.099.

[3] Vikas Dwivedi, Balaji Srinivasan, and Ganapathy Krishnamurthi. Physics informed contour selection for rapid image segmentation. *Scientific Reports*, 14(1):6996, Mar 2024. ISSN 2045-2322. URL https://doi.org/10.1038/s41598-024-57281-x.

[4] Vikas Dwivedi, Bruno Sixou, and Monica Sigovan. Curriculum learning-driven pielms for fluid flow simulations. *Neurocomputing*, 650:130924, 2025. ISSN 0925-2312. URL https://doi.org/10.1016/j.neucom.2025.130924.

[5] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, Jun 2021. ISSN 2522-5820. URL https://doi.org/10.1038/s42254-021-00314-5.

[6] 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, Oct 2024. ISSN 2522-5839. URL https://doi.org/10.1038/s42256-024-00897-5.

[7] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. URL https://doi.org/10.1016/j.jcp.2018.10.045.

[8] Weiwei Zhang, Wei Suo, Jiahao Song, and Wenbo Cao. Physics informed neural networks (pinns) as intelligent computing technique for solving partial differential equations: Limitation and future prospects, 2024. URL https://arxiv.org/abs/2411.18240.

Appendix A. Residual Expressions

- *PDE residual*: For each collocation point within Ω, residuals are calculated from continuity, x and y- momentum equations and set them equal to zero as follows:

$$\boldsymbol{M}\overrightarrow{c} = \overrightarrow{0} \tag{1}$$

where $\boldsymbol{M}$ is a block matrix and $\overrightarrow{c}$ is outer layer weights matrix. Specifically,

$$\begin{bmatrix} M_{u1} & M_{v1} & M_{p1} \\ M_{u2} & M_{v2} & M_{p2} \\ M_{u3} & M_{v3} & M_{p3} \end{bmatrix} \begin{pmatrix} \overrightarrow{c_u} \\ \overrightarrow{c_v} \\ \overrightarrow{c_p} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{2}$$

Here the shape of individual blocks is $1 \times N^*$. The first row corresponds to the continuity equation. The second and third rows correspond to the x and y- momentum equations, respectively. For $k = 1, 2, .., N^*$,

$$M_{u1}(1,k) = -2e^{\xi_k} m_k (m_k x + \alpha_k) \tag{3}$$

$$M_{v1}(1,k) = -2e^{\xi_k} n_k (n_k y + \beta_k) \tag{4}$$

$$M_{p1}(1,k) = 0 \tag{5}$$

$$M_{u2}(1,k) = \hat{u}_{ref} M_{u1}(1,k) + \hat{v}_{ref} M_{v1}(1,k) + \frac{2e^{\xi_k}}{Re}[m_k^2\{1 - 2(m_k x + \alpha_k)^2\} + n_k^2\{1 - 2(n_k y + \beta_k)^2\}] \tag{6}$$

$$M_{v2}(1,k) = 0 \tag{7}$$

$$M_{p2}(1,k) = M_{u1}(1,k) \tag{8}$$

$$M_{u3}(1,k) = 0 \tag{9}$$

$$M_{v3}(1,k) = M_{u2}(1,k) \tag{10}$$

$$M_{p3}(1,k) = M_{v1}(1,k) \tag{11}$$

The terms $\hat{u}_{ref}$ and $\hat{v}_{ref}$ represent the reference velocities used in the quasi-linear approximation of the nonlinear advection terms, as previously discussed.

- *Boundary condition residual:* For each boundary point in $\partial\Omega$, the residual depends on the boundary type (Dirichlet, Neumann, or mixed). For a no-slip condition,

$$\begin{bmatrix} B_{u1} & B_{v1} & B_{p1} \\ B_{u2} & B_{v2} & B_{p2} \end{bmatrix} \begin{bmatrix} \boldsymbol{c_u} \\ \boldsymbol{c_v} \\ \boldsymbol{c_p} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

where for $k = 1, \ldots, N^*$: $B_{u1}(1,k) = e^{\xi_k}$, $B_{v1}(1,k) = B_{p1}(1,k) = 0$, $B_{u2}(1,k) = 0$, $M_{v2}(1,k) = B_{u1}(1,k)$, and $M_{p2}(1,k) = 0$. A horizontal velocity inlet is obtained by replacing the RHS with $\begin{bmatrix} U(y) \\ 0 \end{bmatrix}$.

Appendix B. RBF width distribution

Let the i-th RBF center be $\mathbf{z}_i = (\alpha_i^*, \beta_i^*)$, the special corner be $\mathbf{c} = (c_x, c_y)$, and the boundary-sample set be $\mathcal{B} = \{\mathbf{x}_j\}_{j=1}^{N_{bc}}$. Define

$$d_{\text{corner},i} = \|\mathbf{z}_i - \mathbf{c}\|_2, \qquad d_{\text{bdy},i} = \min_{\mathbf{x}\in\mathcal{B}} \|\mathbf{z}_i - \mathbf{x}\|_2.$$

With constants $r_0 = 0.05$, $\sigma_0 = 0.03$, $L_c > 0$, and $\xi_i \sim \mathcal{N}(0,1)$, the isotropic width parameter σ_i is

$$\sigma_i \;=\; \begin{cases} \sigma_0 + 0.01\,\xi_i, & \text{if } d_{\text{corner},i} < r_0, \\ \min\Big(L_c,\; \sigma_0 + \dfrac{d_{\text{bdy},i}}{L_c}\Big), & \text{otherwise.} \end{cases}$$