

Machine Learning Assisted Two-Level Mixed Finite Element Method for the Darcy–Forchheimer Model

Johnson Daddy Audu[1,3](✉) and Faisal Fairag[2]

[1] Department of Mathematics and Natural Sciences, Prince Mohammed Bin Fahd University, Alkhobar 31952, Saudi Arabia
jaudu@pmu.edu.sa

[2] Department of Mathematics, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
ffairag@kfupm.edu.sa

[3] Center for Cognitive Intelligence & AI Systems, Prince Mohammed Bin Fahd University, Alkhobar 31952, Saudi Arabia

Abstract. We develop and analyse a PINN-assisted two-level mixed finite element method for the Darcy–Forchheimer equation. The coarse nonlinear solve of the classical two-level scheme is replaced by a physics-informed neural network (PINN) surrogate; the network output is projected onto the coarse finite element space and used as the linearization point for the fine-grid correction. Numerical results demonstrate that the proposed approach achieves accuracy comparable to the classical method, with nearly first-order convergence for both velocity and pressure. In multi-query settings where the trained network is reused across repeated solves with fixed physical parameters, the one-time training cost is offset by savings on repeated coarse solves.

Keywords: Darcy–Forchheimer equation · Mixed finite element method · Two-level algorithm · Physics-informed neural networks · Porous media

1 Introduction

Flows through porous media arise in numerous scientific and engineering applications, including petroleum reservoir simulation and groundwater hydrology [3,9]. In classical models, Darcy's law provides a linear relation between the pressure gradient and the velocity field. When the flow velocity is sufficiently large, however, inertial effects become significant and the linear model is no longer adequate [6,13]. The Darcy–Forchheimer equation, which augments Darcy's law with a quadratic drag term, is the standard nonlinear extension used in such regimes.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16766, pp. 236–246, 2027.
https://doi.org/10.1007/978-3-032-30539-8_17

Let $\Omega \subset \mathbb{R}^2$ be a bounded domain with boundary $\partial\Omega$. The steady Darcy–Forchheimer system reads

$$\frac{\mu}{\rho}K^{-1}u + \frac{\beta}{\rho}|u|u + \nabla p = f \quad \text{in } \Omega, \tag{1.1}$$

$$\operatorname{div} u = b \quad \text{in } \Omega, \tag{1.2}$$

$$u \cdot n = g \quad \text{on } \partial\Omega, \tag{1.3}$$

where u and p denote velocity and pressure, K is the permeability tensor, and $\mu, \rho, \beta > 0$ are physical parameters. The nonlinear term $|u|u$ models inertial effects and complicates both the analysis and the numerical solution.

Mixed finite element methods have proved effective for Darcy–Forchheimer-type models, since they approximate velocity and pressure simultaneously and preserve the divergence constraint. Discretization strategies include mixed methods [7,14] and block-centered finite difference methods [11,12]. The resulting discrete systems are nonlinear, however, and these systems grow quickly with mesh refinement.

Two-level methods address this difficulty by solving the nonlinear problem on a coarse mesh and computing a linearized correction on a fine mesh [4]. For the Darcy–Forchheimer model, Fairag and Audu [5] showed that, under the mesh relation $h = O(H^2)$, the two-level scheme achieves the same asymptotic accuracy as a fully nonlinear fine-grid solve while requiring less computational work. The coarse nonlinear problem must still be solved iteratively, however, and this cost remains significant in large-scale or multi-query settings.

Physics-informed neural networks (PINNs) [8,10] offer an alternative by embedding the governing equations into a training loss function. Unlike Newton, the PINN training cost does not grow with mesh refinement, making the reuse argument stronger on larger problems. We propose replacing the coarse nonlinear solve in the two-level scheme with a PINN surrogate. The network is trained to minimize the residual of the Darcy–Forchheimer system, its output is projected onto the coarse finite element space, and the projected field serves as the linearization point for the classical fine-grid correction. We adopt the same P_0–P_1 mixed discretization and linearization strategy as in [5].

We develop a hybrid PINN–finite element framework in which the PINN replaces the coarse nonlinear solver while the fine-grid correction retains the stability of the classical mixed method. We establish a residual-to-error estimate (Theorem 3) bounding the PINN coarse error by a computable indicator η_H^{NN}, and prove global convergence of the full scheme (Theorem 4). Numerical experiments confirm that the PINN-assisted method matches the convergence rates of the classical two-level scheme, and a multi-query experiment shows the training cost is recovered after roughly 47 repeated solves at fixed physical parameters.

Sections 2–5 develop the theory and algorithm; numerical results are in Sect. 6 and conclusions in Sect. 7.

2 Variational Formulation and Analytical Properties

Assume $b \in L^{6/5}(\Omega)$ and $g \in L^{3/2}(\partial\Omega)$ satisfy the compatibility condition

$$\int_\Omega b\,dx = \int_{\partial\Omega} g\,d\sigma. \tag{2.1}$$

Define the function spaces

$$X := L^3(\Omega)^2, \qquad M := W^{1,3/2}(\Omega) \cap L^2_0(\Omega),$$

where $L^2_0(\Omega) = \{q \in L^2(\Omega) : \int_\Omega q\,dx = 0\}$. Since Ω is bounded and $3 > 2$, the embedding $L^3(\Omega) \hookrightarrow L^2(\Omega)$ is continuous, so the $L^2(\Omega)$ duality pairing is well-defined on X.

We define the bilinear form

$$b(v,q) := \int_\Omega \nabla q \cdot v\,dx, \qquad (v,q) \in X \times M, \tag{2.2}$$

and the nonlinear operator

$$\mathcal{A}(v) := \frac{\mu}{\rho} K^{-1} v + \frac{\beta}{\rho}|v|v, \qquad v \in X. \tag{2.3}$$

Let $\kappa_{\min} > 0$ denote the smallest eigenvalue of K. The operator $\mathcal{A}$ satisfies the following properties (see [2,7]):

(i) *Hemicontinuity*: $t \mapsto (\mathcal{A}(v+tw), z)$ is continuous on $[0,1]$ for all $v, w, z \in X$.
(ii) *Monotonicity*: $(\mathcal{A}(v) - \mathcal{A}(w), v - w) \geq \frac{\mu}{\rho}\kappa_{\min}\|v - w\|^2_{L^2}$ for all $v, w \in X$.
(iii) *Continuity*: $\|\mathcal{A}(v) - \mathcal{A}(w)\|_{L^{3/2}} \leq C(1 + \|v\|_{L^3} + \|w\|_{L^3})\|v - w\|_{L^2}$ for all $v, w \in X$.
(iv) *Coercivity*: $(\mathcal{A}(v), v) \geq \frac{\mu}{\rho}\kappa_{\min}\|v\|^2_{L^2}$ for all $v \in X$.
(v) *Inf–sup*: $\inf_{0\neq q\in M} \sup_{0\neq v\in X} \frac{b(v,q)}{\|v\|_{L^3}\|\nabla q\|_{L^{3/2}}} = 1$.

2.1 Variational Formulation

The weak formulation of (1.1)–(1.3) reads: find $(u,p) \in X \times M$ such that

$$(\mathcal{A}(u), \phi) + b(\phi, p) = (f, \phi), \qquad \forall\, \phi \in X, \tag{2.4}$$

$$b(u,q) = -\int_\Omega bq\,dx + \int_{\partial\Omega} gq\,ds, \qquad \forall\, q \in M. \tag{2.5}$$

Under the assumptions $f \in L^{3/2}(\Omega)^2$, $b \in L^{6/5}(\Omega)$, $g \in L^{3/2}(\partial\Omega)$, and the compatibility condition (2.1), properties (ii), (iv), and (v) above together with [7, Theorem 3] guarantee that problem (2.4)–(2.5) admits a unique solution $(u,p) \in X \times M$.

3 Regularization and Linearization

Since $u \mapsto |u|u$ is not differentiable at $u = 0$, we regularize: $|v|_\varepsilon := \sqrt{|v|^2 + \varepsilon^2}$, $F_\varepsilon(v) := |v|_\varepsilon v$, and

$$\mathcal{A}_\varepsilon(v) = \frac{\mu}{\rho} K^{-1} v + \frac{\beta}{\rho} F_\varepsilon(v). \tag{3.1}$$

For fixed $\varepsilon > 0$, F_ε is twice continuously differentiable with Fréchet derivative $DF_\varepsilon(v)w = |v|_\varepsilon w + (v \cdot w/|v|_\varepsilon)v$ and bounded first and second derivatives [5, Lemma 5.1]. Taylor's theorem gives $F_\varepsilon(v) = F_\varepsilon(w) + DF_\varepsilon(w)(v-w) + \mathcal{R}_\varepsilon(v,w)$, where $\|\mathcal{R}_\varepsilon(v,w)\|_{L^2} \le C\|v-w\|_{L^2}^2$. This expansion underpins the linearized fine-grid correction in Sect. 4.

4 Two-Level Discretization

Let $\mathcal{T}_H$ be a shape-regular triangulation of Ω with mesh size H, and let $\mathcal{T}_h$ be a conforming refinement with mesh size $h < H$, obtained by regular refinement of $\mathcal{T}_H$.

4.1 Finite Element Spaces

For $\ell \in \{H, h\}$ define

$$X_\ell := \{v_\ell \in L^3(\Omega)^2 : v_\ell|_K \in P_0(K)^2,\ \forall K \in \mathcal{T}_\ell\},$$
$$M_\ell := \{q_\ell \in C^0(\Omega) \cap W^{1,3/2}(\Omega) : q_\ell|_K \in P_1(K),\ \textstyle\int_\Omega q_\ell\, dx = 0\}.$$

Velocity is approximated by piecewise constants and pressure by continuous piecewise linears. By [14, Proposition 4.1], there exists $\beta_0 > 0$, independent of ℓ, such that

$$\inf_{0 \neq q_\ell \in M_\ell} \sup_{0 \neq v_\ell \in X_\ell} \frac{b(v_\ell, q_\ell)}{\|v_\ell\|_{L^3(\Omega)} \|\nabla q_\ell\|_{L^{3/2}(\Omega)}} \ge \beta_0, \qquad \ell \in \{H, h\}. \tag{4.1}$$

4.2 Classical Two-Level Method

Step 1 (Coarse nonlinear solve). Find $(u_H, p_H) \in X_H \times M_H$ such that

$$(\mathcal{A}(u_H), \phi_H) + b(\phi_H, p_H) = (f, \phi_H), \qquad \forall\, \phi_H \in X_H, \tag{4.2}$$
$$b(u_H, q_H) = -\int_\Omega b q_H\, dx + \int_{\partial\Omega} g q_H\, ds, \quad \forall\, q_H \in M_H. \tag{4.3}$$

The monotonicity and coercivity of $\mathcal{A}$, together with (4.1), guarantee a unique solution (u_H, p_H) (see [7, Theorem 3]).

Step 2 (Fine-Grid Linearized Correction). The coarse velocity u_H is prolonged to $\mathcal{T}_h$ (still denoted u_H) and used as the linearization point in the regularized formulation. Find $(u_h, p_h) \in X_h \times M_h$ such that

$$(D\mathcal{A}_\varepsilon(u_H)u_h, \phi_h) + b(\phi_h, p_h) = (f, \phi_h) + (D\mathcal{A}_\varepsilon(u_H)u_H - \mathcal{A}_\varepsilon(u_H), \phi_h), \quad \forall\, \phi_h \in X_h, \tag{4.4}$$

$$b(u_h, q_h) = -\int_\Omega b q_h \, dx + \int_{\partial\Omega} g q_h \, ds, \qquad \forall\, q_h \in M_h. \tag{4.5}$$

Coercivity of $D\mathcal{A}_\varepsilon(u_H)$ and the inf–sup condition (4.1) ensure unique solvability of the linear system (4.4)–(4.5) [5, Section 5.2].

4.3 Error Estimates for the Classical Scheme

The following estimates are drawn from [1,5].

Theorem 1. *Let (u_H, p_H) solve (4.2)–(4.3) and (u_h^{NL}, p_h^{NL}) the fully nonlinear fine-grid problem. There exist constants $C > 0$, independent of H and h, such that*

$$\|u - u_H\|_{L^2} + \|\nabla(p - p_H)\|_{L^{3/2}} \le CH, \tag{4.6}$$

$$\|u - u_h^{NL}\|_{L^2} + \|\nabla(p - p_h^{NL})\|_{L^{3/2}} \le Ch. \tag{4.7}$$

Theorem 2. (Two-level estimate [5]) *Let (u_h, p_h) be the two-level approximation from (4.4)–(4.5). There exists $C > 0$, independent of h and H, such that*

$$\|u - u_h\|_{L^2} + \|\nabla(p - p_h)\|_{L^{3/2}} \le C(\varepsilon h + h + H^2 + \varepsilon). \tag{4.8}$$

Under $h = H^2$, this matches the fully nonlinear fine-grid rate (4.7).

5 PINN-Assisted Two-Level Method

We extend the classical scheme by replacing the coarse nonlinear solve with a PINN surrogate. The fine-grid correction is unchanged.

5.1 Algorithm

Step 1 (PINN Coarse Surrogate). Let $(\hat{u}(\cdot;\Theta), \hat{p}(\cdot;\Theta))$ be neural network approximations of the velocity and pressure fields with weights Θ. Define the equation residuals

$$r_{\text{mom}} = \mathcal{A}(\hat{u}) + \nabla\hat{p} - f, \tag{5.1}$$

$$r_{\text{div}} = \operatorname{div}\hat{u} - b, \tag{5.2}$$

$$r_{\text{bc}} = \hat{u}\cdot n - g, \tag{5.3}$$

and the physics-informed loss

$$J(\Theta) = \omega_1\|r_{\text{mom}}\|^2_{L^2(\Omega)} + \omega_2\|r_{\text{div}}\|^2_{L^2(\Omega)} + \omega_3\|r_{\text{bc}}\|^2_{L^2(\partial\Omega)}. \tag{5.4}$$

After training to obtain Θ^*, set $\hat{u} = \hat{u}(\cdot;\Theta^*)$ and $\hat{p} = \hat{p}(\cdot;\Theta^*)$, and enforce zero mean pressure via $\hat{p} \leftarrow \hat{p} - |\Omega|^{-1}\int_\Omega \hat{p}\, dx$.

Step 2 (Projection onto the Coarse Finite Element Space). Let $\Pi_H^u : L^2(\Omega)^2 \to X_H$ and $\Pi_H^p : W^{1,3/2}(\Omega) \to M_H$ be stable L^2-projections. Define

$$u_H^{NN} = \Pi_H^u \hat{u}, \qquad p_H^{NN} = \Pi_H^p \hat{p}. \tag{5.5}$$

Step 3 (Surrogate Quality Check). Before proceeding to the fine-grid solve, we measure how well the projected PINN pair satisfies the discrete equations. Define the coarse residual functional

$$R_H^{NN}(\phi, q) = (f, \phi) - (\mathcal{A}(u_H^{NN}), \phi) - b(\phi, p_H^{NN}) - b(u_H^{NN}, q) - \int_\Omega bq\, dx + \int_{\partial\Omega} gq\, ds, \tag{5.6}$$

and its dual norm

$$\eta_H^{NN} = \sup_{\phi \in X,\ q \in M(\phi,q) \neq (0,0)} \frac{|R_H^{NN}(\phi, q)|}{\|\phi\|_{L^3(\Omega)} + \|\nabla q\|_{L^{3/2}(\Omega)}}. \tag{5.7}$$

The supremum in (5.7) is over an infinite-dimensional space and is not evaluated numerically in this work; η_H^{NN} instead serves as the theoretical quantity through which Theorem 3 certifies the surrogate.

Step 4 (Fine-Grid Linearized Correction). Solve the linearized system (4.4)–(4.5) with u_H replaced by u_H^{NN} to obtain $(u_h, p_h) \in X_h \times M_h$. By Theorem 3, a small η_H^{NN} ensures the PINN coarse error is small, and Theorem 4 then controls the resulting fine-mesh error via (5.9).

The complete procedure is summarized in Algorithm 1.

Algorithm 1. PINN-Assisted Two-Level Mixed Finite Element Method

1: **Input:** Domain Ω; mesh sizes H, h; parameters μ, ρ, β, K; data (f, b, g).
2: Train PINN: minimize $J(\Theta)$ in (5.4).
3: Project: $u_H^{NN} = \Pi_H^u \hat{u}$, $p_H^{NN} = \Pi_H^p \hat{p}$.
4: Solve the linearized fine-grid system (4.4)–(4.5) with u_H replaced by u_H^{NN}.
5: **Output:** (u_h, p_h).

Remark 1. By Theorem 3, η_H^{NN} bounds the coarse approximation error without the exact solution, so it can be inspected before the fine solve to gauge how much the result should be trusted. In the present scheme the fine solve always proceeds; an adaptive variant that triggers retraining when η_H^{NN} is large is left for future work.

Theorem 3. *Let $(u, p) \in X \times M$ solve (2.4)–(2.5), and let $(u_H^{NN}, p_H^{NN}) \in X_H \times M_H$ be defined by (5.5). Assume (4.1) holds. There exists $C > 0$, independent of H, such that*

$$\begin{aligned} &\|u - u_H^{NN}\|_{L^2(\Omega)} + \|\nabla(p - p_H^{NN})\|_{L^{3/2}(\Omega)} \\ &\quad \le C\Big(\inf_{v_H \in X_H} \|u - v_H\|_{L^2(\Omega)} + \inf_{q_H \in M_H} \|\nabla(p - q_H)\|_{L^{3/2}(\Omega)} + \eta_H^{NN} \Big). \end{aligned} \tag{5.8}$$

Proof. (Proof sketch) Set $e_u = u - u_H^{NN}$ and $e_p = p - p_H^{NN}$. Subtracting (5.6) from (2.4)–(2.5) gives $(\mathcal{A}(u)-\mathcal{A}(u_H^{NN}),\phi)+b(\phi,e_p)+b(e_u,q) = R_H^{NN}(\phi,q)$. Testing with $(e_u,-e_p)$ and using monotonicity (ii) yields $c_0\|e_u\|_{L^2}^2 \le \eta_H^{NN}(\|e_u\|_{L^3} + \|\nabla e_p\|_{L^{3/2}})$. The inf–sup condition (v) and continuity (iii) bound $\|\nabla e_p\|_{L^{3/2}}$, and combining gives $\|e_u\|_{L^2} + \|\nabla e_p\|_{L^{3/2}} \le C\eta_H^{NN}$. The full estimate (5.8) follows by the triangle inequality.

5.2 Global Convergence of the PINN-Assisted Scheme

Theorem 4. *Let $(u,p) \in X \times M$ be the exact solution of (2.4)–(2.5), and let (u_h,p_h) be the PINN-assisted two-level approximation. Let (u_h^{NL},p_h^{NL}) denote the fully nonlinear fine-grid solution and (u_h^{TL},p_h^{TL}) the classical two-level approximation linearized about u_H. There exists $C>0$, independent of h, H, and η_H^{NN}, such that*

$$\|u-u_h\|_{L^2(\Omega)} + \|\nabla(p-p_h)\|_{L^{3/2}(\Omega)} \le C\big(\varepsilon h + h + H^2 + \varepsilon + H + \eta_H^{NN}\big). \quad (5.9)$$

Proof. (Proof sketch) Decompose $u-u_h = (u-u_h^{NL})+(u_h^{NL}-u_h^{TL})+(u_h^{TL}-u_h)$ and bound each term. The first two terms are bounded by Theorems 1 and 2. For the third, subtracting the PINN-assisted system from the classical two-level system and using Lipschitz continuity of $D\mathcal{A}_\varepsilon$ gives $T_3 \le C\|u_H^{NN} - u_H\|_{L^2}$. The triangle inequality and Theorems 1 and 3 then yield $\|u_H^{NN} - u_H\|_{L^2} \le C(H+\eta_H^{NN})$, which proves (5.9).

Under $h = H^2$ and $\eta_H^{NN} = O(H)$, estimate (5.9) gives first-order convergence in h, matching the classical two-level rate.

6 Numerical Experiments

All computations use the manufactured exact solution

$$u(x,y) = \begin{pmatrix} \sin(\pi x)\sin(\pi y) \\ \cos(\pi x)\sin(\pi y) \end{pmatrix}, \qquad p(x,y) = \sin(\pi x)\cos(\pi y), \quad (6.1)$$

on $\Omega = (0,1)^2$. The source term f, divergence data b, and boundary data $g = u\cdot n$ are derived from (1.1)–(1.3) so that (6.1) satisfies the system exactly. Unless stated otherwise, $\mu = \rho = \beta = 1$, $K = I$, and $\varepsilon = 10^{-6}$.

The PINN uses a fully connected network with three hidden layers of 40 neurons and hyperbolic tangent activations. It is trained by minimizing $J(\Theta)$ in (5.4) with equal weights $\omega_1 = \omega_2 = \omega_3 = 1$, using the Adam optimizer at learning rate 10^{-3} for 1200 epochs, with 800 interior and 300 boundary collocation points per batch. All computations were carried out in MATLAB ONLINE R2025b.

Table 1. One-grid mixed FEM errors.

H	$\|u - u_H\|_{L^2}$	Rate	$\|\nabla(p - p_H)\|_{L^{3/2}}$	Rate
1/8	1.8646e−01	–	4.0750e−01	–
1/16	9.5546e−02	0.9646	2.0604e−01	0.9839
1/32	4.8157e−02	0.9884	1.0328e−01	0.9963
1/64	2.4138e−02	0.9964	5.1674e−02	0.9991
1/128	1.2069e−02	0.9980	2.5840e−02	0.9990

Table 2. Classical two-level method errors ($h = H^2$).

H	h	$\|u - u_h\|_{L^2}$	Rate_u	$\|\nabla(p - p_h)\|_{L^{3/2}}$	Rate_p
1/2	1/4	3.698553e−01	–	7.909967e−01	–
1/4	1/16	1.032088e−01	0.9207	2.216191e−01	0.9178
1/8	1/64	2.648289e−02	0.9812	5.704493e−02	0.9790
1/16	1/256	6.707819e−03	0.9906	1.442337e−02	0.9918

6.1 Numerical Results

Table 1 verifies first-order convergence of the base discretization. Both two-level methods under $h = H^2$ (Tables 2 and 3) achieve nearly first-order convergence in h, consistent with Theorems 2 and 4. Figure 1 confirms the slopes are indistinguishable.

6.2 Effect of the Forchheimer Number

To assess robustness under stronger nonlinearity, we fix $H = 1/8$, $h = 1/64$ and vary $\beta \in \{1, 5, 10, 20\}$, comparing both methods at each value. The results are reported in Table 4.

At $\beta = 1$ both methods agree closely. As β increases, the PINN pressure error grows, by a factor of roughly 13 between $\beta = 1$ and $\beta = 20$, against a factor of about 3 for the classical scheme. The benefit of pressure prolongation diminishes at higher β, since the quality of p_H^{NN} is bounded by that of the underlying surrogate.

Table 3. PINN-assisted two-level method errors ($h = H^2$).

H	h	$\|u - u_h\|_{L^2}$	Rate_u	$\|\nabla(p - p_h)\|_{L^{3/2}}$	Rate_p
1/2	1/4	3.459306e−01	–	7.830036e−01	–
1/4	1/16	9.560424e−02	0.9277	2.083135e−01	0.9551
1/8	1/64	2.429132e−02	0.9883	5.238011e−02	0.9958
1/16	1/256	6.267570e−03	0.9772	1.358593e−02	0.9735

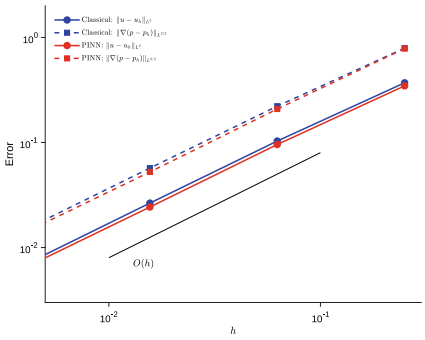

Fig. 1. Log–log convergence of both methods. Both follow the $O(h)$ slope.

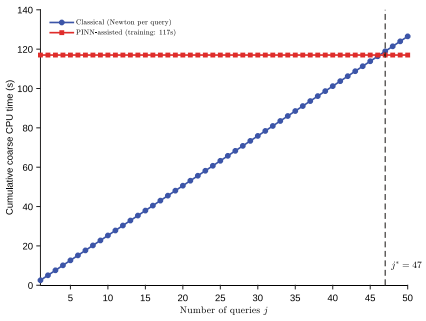

Fig. 2. Cumulative coarse CPU time at $H = 1/128$, $\beta = 20$. PINN becomes cheaper after $j^* = 47$ queries.

Table 4. Errors at fixed $H = 1/8$, $h = 1/64$ for varying β.

	Classical		PINN-assisted	
β	$\|u - u_h\|_{L^2}$	$\|\nabla(p - p_h)\|_{L^{3/2}}$	$\|u - u_h\|_{L^2}$	$\|\nabla(p - p_h)\|_{L^{3/2}}$
1	2.6483e−02	5.7045e−02	2.4373e−02	5.2486e−02
5	1.9430e−02	7.4865e−02	1.9212e−02	8.0660e−02
10	1.7977e−02	1.0034e−01	2.3777e−02	1.7758e−01
20	1.7304e−02	1.5820e−01	2.6188e−02	6.7849e−01

6.3 Computational Cost

Table 5 reports CPU times for both methods. PINN training costs roughly 26–28 s at every mesh size, while Newton on the coarse mesh takes 15–48 ms. The fine-grid times are close—both methods solve the same linearized system—though at the finest mesh the PINN fine solve is slightly faster (118s vs 124s). In a single run the PINN-assisted method is more expensive; Sect. 6.4 shows when this reverses.

6.4 Multi-query Reuse

To test the reuse argument, we fix $H = 1/128$, $\beta = 20$, and train the PINN once on the base manufactured solution. We then run 50 queries with varying source frequency $j = 1, 2, \ldots, 50$, where each query uses the manufactured solution

$$u^{(j)}(x, y) = \begin{pmatrix} \sin(j\pi x)\sin(\pi y) \\ \cos(j\pi x)\sin(\pi y) \end{pmatrix}, \quad p^{(j)}(x, y) = \sin(j\pi x)\cos(\pi y),$$

giving a different source term $f^{(j)}$ while keeping β, K, μ, ρ fixed. For each query, the classical method pays a full Newton solve on the coarse mesh; the

Table 5. CPU time comparison (seconds).

H	h	Method	Coarse (s)	Fine (s)	Total (s)
1/4	1/16	Classical	1.91e−02	4.34e−02	6.26e−02
1/4	1/16	PINN-assisted	2.76e+01	2.98e−02	2.76e+01
1/8	1/64	Classical	1.52e−02	1.57e+00	1.58e+00
1/8	1/64	PINN-assisted	2.62e+01	1.59e+00	2.78e+01
1/16	1/256	Classical	4.77e−02	1.24e+02	1.25e+02
1/16	1/256	PINN-assisted	2.67e+01	1.18e+02	1.45e+02

PINN method pays nothing at the coarse stage since the network is already trained. The fine-grid cost is identical for both methods and does not affect the comparison.

Training took 117 s, against an average Newton time of 2.53 s per query. The PINN cumulative cost crosses below the classical at query $j^* = 47$, in close agreement with the theoretical estimate $t_{\text{train}}/t_{\text{Newton}} \approx 46$. Figure 2 shows the cumulative coarse CPU time for both methods as a function of query number.

7 Conclusion

We introduced a PINN-assisted two-level mixed finite element method for the Darcy–Forchheimer equation. The PINN replaces the coarse nonlinear solve; its output is projected onto the coarse FE space and used to linearize the fine-grid problem. Theorem 3 bounds the PINN coarse error via η_H^{NN}, and Theorem 4 establishes first-order convergence under $h = H^2$. The method matches the classical scheme at $\beta = 1$; pressure errors grow at higher β. A multi-query experiment shows the PINN becomes cheaper after $j^* = 47$ repeated solves.

References

1. Audu, J.D., Fairag, F., Mustapha, K.: Mixed finite element analysis for generalized Darcy-Forchheimer model in porous media. J. Comput. Appl. Math. **353**, 191–203 (2019)
2. Audu, J.D., Fairag, F.A., Messaoudi, S.A.: On the well-posedness of generalized Darcy-Forchheimer equation. Bound. Value Probl., 123 (2018)
3. Bear, J.: Dynamics of Fluids in Porous Media. Elsevier (1972)
4. Fairag, F.: Two level finite element technique for pressure recovery from stream function formulation of the Navier-Stokes equations. In: Barth, T.J., Chan, T., Haimes, R. (eds.) Multiscale and Multiresolution Methods, pp. 297–306. Springer (2002)
5. Fairag, F., Audu, J.: Two-level Galerkin mixed finite element method for the Darcy-Forchheimer model in porous media. SIAM J. Numer. Anal. **59**(2), 888–911 (2021)
6. Forchheimer, P.: Wasserbewegung durch Boden. Zeitschrift des Vereines Deutscher Ingenieure **45**, 1782–1788 (1901)

7. Girault, V., Wheeler, M.F.: Numerical discretization of a Darcy-Forchheimer model. Numer. Math. **110**(2), 161–198 (2008)
8. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L.: Physics-informed machine learning. Nat. Rev. Phys. **3**, 422–440 (2021)
9. Muskat, M.: The Flow of Homogeneous Fluids Through Porous Media. McGraw-Hill (1946)
10. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
11. Rui, H., Pan, H.: A block-centered finite difference method for the Darcy-Forchheimer model. SIAM J. Numer. Anal. **50**(5), 2612–2631 (2012)
12. Rui, H., Zhao, D., Pan, H.: A block-centered finite difference method for Darcy-Forchheimer model with variable Forchheimer number. Numer. Methods Partial Differ. Equ. **31**, 1603–1622 (2015)
13. Ruth, D., Ma, H.: On the derivation of the Forchheimer equation by means of the averaging theorem. Transp. Porous Media **7**(3), 255–264 (1992)
14. Salas, J.J., López, H., Molina, B.: An analysis of a mixed finite element method for a Darcy-Forchheimer model. Math. Comput. Model. **57**(9), 2325–2338 (2013)