# LM-PIELM: A Lagrange-Multiplier Framework for Hard Constraint Enforcement in Physics-Informed Extreme Learning Machines for PDE-Governed Forward and Inverse Problems

**Eshan Kulkarni** [*1] **Rishi Mishra** [*2] **Balaji Srinivasan** [1]

## Abstract

Physics-Informed Extreme Learning Machines (PIELMs) solve PDEs as linear least-squares problems by fixing the feature layer and learning only the output weights, but boundary conditions are usually imposed through soft penalties with problem-dependent weights. We propose LM-PIELM, a Lagrange-multiplier formulation that imposes boundary conditions as algebraic equality constraints using the linear-in-parameters structure of PIELMs. The resulting constrained least-squares problem leads to a linear saddle-point system with no boundary penalty tuning, no iterative dual updates, and no problem-specific analytical constrained ansatz. An equivalent augmented-system implementation avoids explicit normal-equation Gram matrices and improves robustness in ill-conditioned regimes. Since the constraint block uses only boundary operator evaluations, the same formulation applies to mixed boundary conditions, irregular geometries, and inverse problems including parameter identification and source recovery from noisy data. Across forward and inverse benchmarks, the augmented-system form preserves small boundary residuals, improves solution accuracy in ill-conditioned regimes, and keeps boundary admissibility essentially decoupled from sensor count, noise level, and reconstruction quality.

[1]Wadhwani School of Data Science & AI, Indian Institute of Technology Madras, Chennai, Tamil Nadu, India [2]Department of Engineering Design, Indian Institute of Technology Madras, Chennai, Tamil Nadu, India. Correspondence to: Rishi Mishra <ed20d006@smail.iitm.ac.in>, Eshan Kulkarni <me21b057@smail.iitm.ac.in>.

*Proceedings of the AI4Physics Workshop at the $43^{rd}$ International Conference on Machine Learning (AI4Physics@ICML 2026)*, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

## 1. Introduction

The numerical solution of partial differential equations (PDEs) is central to computational science and engineering. Classical mesh-based methods such as finite elements and finite differences achieve high accuracy but require structured or semi-structured discretizations that can be costly to construct for complex geometries. The past decade has seen the emergence of *physics-informed* machine learning methods that replace traditional discretizations with neural-network or random-feature representations, embedding the governing equations directly into the learning objective (Lagaris et al., 1998; Raissi et al., 2019).

Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019) embed PDE residuals, boundary conditions, and initial conditions into a unified loss function, offering a mesh-free alternative to classical solvers with demonstrated success across a range of scientific and engineering problems (Karniadakis et al., 2021; Cuomo et al., 2022). Physics-Informed Extreme Learning Machines (PIELMs) (Dwivedi & Srinivasan, 2020) extend this idea through the Extreme Learning Machine (ELM) architecture (Huang et al., 2006), in which the feature layer is fixed and only the output weights are trained. While standard ELMs often use randomly generated hidden features, ELM-type solvers based on deterministic polynomial features such as Legendre, Bernstein, and Chebyshev bases have also been used for differential equations (Yang et al., 2018; Sun et al., 2019; Liu et al., 2020; Yang et al., 2020; Liu et al., 2021). This linear-in-parameters structure reduces the nonlinear PINN training problem to a linear least-squares solve, eliminating non-convex optimization while retaining the approximation power required to solve PDEs accurately.

Despite these advantages, both PINNs and PIELMs commonly enforce boundary conditions through soft penalty terms added to a composite loss function. Physical constraints are satisfied only approximately, with accuracy governed by user-specified penalty weights. The magnitudes of different loss components often differ by several orders of magnitude, leading the optimizer to reduce the interior residual at the expense of boundary accuracy (Wang et al., 2021;

Krishnapriyan et al., 2021). Adaptive reweighting strategies (Xiang et al., 2022) mitigate but do not remove this limitation as constraint satisfaction remains approximate, and boundary errors can propagate into the interior solution, effectively solving a different boundary value problem than the one intended.

Several approaches have attempted to enforce boundary conditions exactly. Distance-function methods (Lagaris et al., 1998; Sukumar & Srivastava, 2022) multiply the network output by a function that vanishes on the boundary, guaranteeing Dirichlet satisfaction by construction, but constructing suitable distance functions becomes difficult for irregular geometries or Neumann conditions. The Theory of Functional Connections (TFC) and its extreme learning variant X-TFC (Mortari, 2017; Schiassi et al., 2021) achieve near-machine-precision enforcement by embedding constraints analytically into the solution representation, but the required constrained expressions are derived per problem class and do not extend naturally to irregular geometries, mixed boundary conditions, or inverse settings. In a related direction, (Mishra et al., 2026) enforce boundary conditions exactly via algebraic projection within a PIELM framework for eigenvalue problems, demonstrating that exact discrete enforcement is both feasible and beneficial in the ELM setting.

Augmented Lagrangian PINNs (AL-PINNs) (Son et al., 2023) introduce Lagrange multipliers within the PINN framework, but because they operate inside the nonlinear training loop, they still require iterative dual-variable updates and augmented penalty terms for stability. Dong & Ni (2021) propose a penalty-free Lagrange multiplier formulation for fully trainable PINNs. However, the nonlinearity of the PINN model still requires iterative solution of the resulting KKT system. Variational formulations such as the Deep Ritz method (E & Yu, 2018) and hp-VPINNs (Kharazmi et al., 2021) improve training through a weak form but still leave boundary conditions softly enforced. The constrained formulation used here is also related to classical equality-constrained least-squares and augmented-system formulations in numerical linear algebra (Golub & Van Loan, 2013; Benzi et al., 2005), applied in this work to the physics-informed fixed-feature setting.

In this work, we propose LM-PIELM, a framework that enforces boundary conditions as hard algebraic constraints within the fixed-feature PIELM setting. By introducing Lagrange multipliers, the learning problem is reformulated as an equality-constrained least-squares problem whose KKT conditions give a structured linear saddle-point system. This system can be solved directly without boundary penalty-weight tuning, iterative dual updates, or a problem-specific analytical constrained ansatz. We also use an equivalent augmented-system implementation, which introduces the PDE residual as an auxiliary unknown and avoids explicit formation of the normal-equation Gram matrix. The Lagrange constraint block requires only boundary operator evaluations at collocation points, so the same algebraic structure applies across domain geometries, boundary condition types, and inverse problem settings. We make the following contributions:

- We develop a Lagrange-multiplier formulation of PIELM in which boundary conditions are imposed as algebraic equality constraints, leading to a linear saddle-point system for the constrained forward solve.
- We present an augmented-system implementation of the same constrained least-squares problem, avoiding explicit normal-equation Gram matrices and improving robustness in ill-conditioned regimes.
- We show that the formulation extends without structural change to mixed Dirichlet-Neumann conditions, irregular domains, and inverse problems including scalar parameter identification and source recovery.
- We validate the framework on five main benchmarks spanning forward and inverse problems, demonstrating that boundary constraints are preserved at the discrete level throughout. Additional experiments, including an unsteady 1D advection-diffusion problem in a space-time setting and two further inverse benchmarks, are reported in Appendices A and B.

## 2. LM-PIELM: Lagrange-Multiplier Framework for Hard Boundary Enforcement

### 2.1. Physics-Informed Extreme Learning Machines

We consider a linear boundary value problem of the form

$$\mathcal{L}u(\mathbf{x}) = f(\mathbf{x}), \qquad \mathbf{x} \in \Omega, \tag{1}$$

together with boundary conditions

$$\mathcal{B}_m u(\mathbf{x}) = g_m(\mathbf{x}), \qquad \mathbf{x} \in \Gamma_m, \quad m = 1, \ldots, M_b, \tag{2}$$

where $\partial\Omega = \bigcup_{m=1}^{M_b} \Gamma_m$ is a partition of the boundary into segments that may carry different boundary operators.

In a Physics-Informed Extreme Learning Machine (PIELM), the solution is approximated by a single-hidden-layer model with fixed basis functions (Huang et al., 2006; Dwivedi & Srinivasan, 2020):

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{N} C_j \phi_j(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{C}, \tag{3}$$

where $\{\phi_j\}_{j=1}^{N}$ are fixed nonlinear features and $\mathbf{C} \in \mathbb{R}^N$ is the vector of output coefficients. The important point is that $\hat{u}$ is linear in $\mathbf{C}$. As a result, applying the PDE operator or a boundary operator to $\hat{u}$ also remains linear in $\mathbf{C}$.

Let $\{\mathbf{x}_i^c\}_{i=1}^{N_c} \subset \Omega$ be interior collocation points and let $\{\mathbf{x}_k^b\}_{k=1}^{N_b} \subset \partial\Omega$ be boundary collocation points. To keep the notation compact, we write $\mathcal{B}_k$ for the boundary operator associated with the $k$th boundary point, so $\mathcal{B}_k$ may represent either a Dirichlet row, a Neumann row, or any other linear boundary condition prescribed there. We then assemble

$$\Psi_{ij} = \mathcal{L}\phi_j(\mathbf{x}_i^c), \qquad f_i = f(\mathbf{x}_i^c), \tag{4}$$

and

$$(\Phi_b)_{kj} = \mathcal{B}_k\phi_j(\mathbf{x}_k^b), \qquad g_k = g(\mathbf{x}_k^b). \tag{5}$$

The standard PIELM treatment enforces the PDE and boundary conditions through a penalized least-squares objective,

$$\min_{\mathbf{C}} \frac{1}{2}\|\mathbf{\Psi C}-\mathbf{f}\|_2^2+\frac{\mu_{\mathrm{BC}}}{2}\|\mathbf{\Phi}_b\mathbf{C}-\mathbf{g}\|_2^2, \qquad \mu_{\mathrm{BC}} > 0, \tag{6}$$

which yields the normal equations

$$\left(\mathbf{\Psi}^\top\mathbf{\Psi} + \mu_{\mathrm{BC}}\,\mathbf{\Phi}_b^\top\mathbf{\Phi}_b\right)\mathbf{C} = \mathbf{\Psi}^\top\mathbf{f} + \mu_{\mathrm{BC}}\,\mathbf{\Phi}_b^\top\mathbf{g}. \tag{7}$$

### 2.2. Equality-Constrained LM-PIELM Formulation

Equation (7) shows the basic limitation of the penalty formulation. The interior PDE residual and the boundary residual are balanced through a user-chosen weight $\mu_{\mathrm{BC}}$. This formulation is convenient but enforces boundary conditions only softly. A large $\mu_{\mathrm{BC}}$ improves boundary satisfaction at the cost of ill conditioning the system and degrading interior accuracy. A small $\mu_{\mathrm{BC}}$ preserves conditioning but allows boundary errors to persist and propagate inward. The optimal value is problem dependent and unknown a priori.

To eliminate this trade-off, we reformulate the PIELM problem as an equality-constrained least-squares problem:

$$\min_{\mathbf{C}} \frac{1}{2}\|\mathbf{\Psi C} - \mathbf{f}\|_2^2 \quad \text{subject to} \quad \mathbf{\Phi}_b\mathbf{C} = \mathbf{g}. \tag{8}$$

This retains the PDE residual in least-squares form while imposing the boundary conditions algebraically at the enforced boundary collocation points. No boundary penalty weight is required.

Introducing Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^{N_b}$ for the equality constraints in (8), the Lagrangian is

$$L(\mathbf{C}, \boldsymbol{\lambda}) = \frac{1}{2}\|\mathbf{\Psi C} - \mathbf{f}\|_2^2 + \boldsymbol{\lambda}^\top(\mathbf{\Phi}_b\mathbf{C} - \mathbf{g}). \tag{9}$$

The first-order optimality conditions are

$$\nabla_{\mathbf{C}} L = \mathbf{\Psi}^\top(\mathbf{\Psi C} - \mathbf{f}) + \mathbf{\Phi}_b^\top\boldsymbol{\lambda} = \mathbf{0}, \tag{10}$$

$$\nabla_{\boldsymbol{\lambda}} L = \mathbf{\Phi}_b\mathbf{C} - \mathbf{g} = \mathbf{0}. \tag{11}$$

Assembling (10)-(11) into a single block system gives the LM-PIELM saddle-point system:

$$\underbrace{\begin{bmatrix} \mathbf{\Psi}^\top\mathbf{\Psi} & \mathbf{\Phi}_b^\top \\ \mathbf{\Phi}_b & \mathbf{0} \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} \mathbf{C} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{\Psi}^\top\mathbf{f} \\ \mathbf{g} \end{bmatrix}. \tag{12}$$

Solving (12) yields output weights $\mathbf{C}$ that minimize the interior PDE residual while satisfying the discrete boundary constraints. The multipliers $\boldsymbol{\lambda}$ are dual variables associated with the boundary constraints. Because the objective in (8) is quadratic and the constraints are linear, the KKT system (12) is a linear system, solvable by standard dense linear algebra in a single, non-iterative step. The linear block structure is a direct consequence of the fixed-feature PIELM ansatz. In contrast, applying the same multiplier construction to a fully trainable neural network would produce a nonlinear KKT system requiring iterative solution (Son et al., 2023; Dong & Ni, 2021).

The same algebraic framework covers Dirichlet, Neumann, and mixed boundary conditions without changing the structure of the system. For a Dirichlet condition, a row of $\mathbf{\Phi}_b$ contains pointwise evaluations $\phi_j(\mathbf{x}_k^b)$. For a Neumann condition $\partial u/\partial n = h$ at $\mathbf{x}_k^b$, the corresponding row is

$$(\Phi_b)_{kj} = \nabla\phi_j(\mathbf{x}_k^b) \cdot \mathbf{n}(\mathbf{x}_k^b). \tag{13}$$

Thus mixed boundary conditions are handled simply by stacking the appropriate boundary rows into the same matrix $\mathbf{\Phi}_b$ and the prescribed values into the same vector $\mathbf{g}$. Nothing in the derivation depends on a special geometry, a special boundary type, or a problem-specific constrained ansatz.

### 2.3. Augmented-System Saddle-Point Implementation

The reduced saddle-point system (12) contains the normal-equation matrix $\mathbf{\Psi}^\top\mathbf{\Psi}$. Although this form is compact, explicitly forming this Gram matrix can amplify conditioning effects associated with $\mathbf{\Psi}$. To avoid the normal-equation form, we use the classical augmented-system formulation for least-squares problems (Golub & Van Loan, 2013; Benzi et al., 2005).

We introduce the auxiliary residual variable

$$\mathbf{r} = \mathbf{\Psi C} - \mathbf{f} \in \mathbb{R}^{N_c}. \tag{14}$$

Then

$$\frac{1}{2}\|\mathbf{\Psi C} - \mathbf{f}\|_2^2 = \frac{1}{2}\mathbf{r}^\top\mathbf{r}, \tag{15}$$

provided that the residual definition $\mathbf{r} - \mathbf{\Psi C} + \mathbf{f} = \mathbf{0}$ is

enforced. Therefore, (8) can be equivalently written as

$$\begin{aligned} \min_{\mathbf{r},\mathbf{C}} \quad & \frac{1}{2}\mathbf{r}^\top \mathbf{r}, \\ \text{subject to} \quad & \mathbf{r} - \mathbf{\Psi}\mathbf{C} + \mathbf{f} = \mathbf{0}, \\ & \mathbf{\Phi}_b \mathbf{C} = \mathbf{g}. \end{aligned} \tag{16}$$

Introducing multipliers $\boldsymbol{\gamma}$ and $\boldsymbol{\lambda}$ for the residual-definition constraint and the boundary constraint, respectively, gives

$$\begin{aligned} L(\mathbf{r}, \mathbf{C}, \boldsymbol{\gamma}, \boldsymbol{\lambda}) = & \frac{1}{2}\mathbf{r}^\top \mathbf{r} + \boldsymbol{\gamma}^\top (\mathbf{r} - \mathbf{\Psi}\mathbf{C} + \mathbf{f}) \\ & + \boldsymbol{\lambda}^\top (\mathbf{\Phi}_b \mathbf{C} - \mathbf{g}). \end{aligned} \tag{17}$$

The stationarity conditions are

$$\frac{\partial L}{\partial \mathbf{r}} = \mathbf{r} + \boldsymbol{\gamma} = \mathbf{0} \quad \Rightarrow \quad \boldsymbol{\gamma} = -\mathbf{r}, \tag{18a}$$

$$\frac{\partial L}{\partial \mathbf{C}} = -\mathbf{\Psi}^\top \boldsymbol{\gamma} + \mathbf{\Phi}_b^\top \boldsymbol{\lambda} = \mathbf{0} \quad \Rightarrow \quad \mathbf{\Psi}^\top \mathbf{r} + \mathbf{\Phi}_b^\top \boldsymbol{\lambda} = \mathbf{0}, \tag{18b}$$

$$\frac{\partial L}{\partial \boldsymbol{\lambda}} = \mathbf{\Phi}_b \mathbf{C} - \mathbf{g} = \mathbf{0}, \tag{18c}$$

$$\frac{\partial L}{\partial \boldsymbol{\gamma}} = \mathbf{r} - \mathbf{\Psi}\mathbf{C} + \mathbf{f} = \mathbf{0}. \tag{18d}$$

These optimality conditions give the equivalent first-order system

$$\mathbf{r} - \mathbf{\Psi}\mathbf{C} = -\mathbf{f}, \tag{19a}$$

$$\mathbf{\Psi}^\top \mathbf{r} + \mathbf{\Phi}_b^\top \boldsymbol{\lambda} = \mathbf{0}, \tag{19b}$$

$$\mathbf{\Phi}_b \mathbf{C} = \mathbf{g}. \tag{19c}$$

Assembling (19a)-(19c) into a single block system yields the augmented-system saddle-point formulation:

$$\underbrace{\begin{bmatrix} \mathbf{I}_{N_c} & -\mathbf{\Psi} & \mathbf{0} \\ \mathbf{\Psi}^\top & \mathbf{0} & \mathbf{\Phi}_b^\top \\ \mathbf{0} & \mathbf{\Phi}_b & \mathbf{0} \end{bmatrix}}_{\mathbf{K}_{\text{aug}}} \begin{bmatrix} \mathbf{r} \\ \mathbf{C} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{f} \\ \mathbf{0} \\ \mathbf{g} \end{bmatrix}. \tag{20}$$

Eliminating $\mathbf{r}$ from (20) recovers the reduced system (12). Hence the two systems are equivalent in exact arithmetic. The difference is numerical. The augmented-system form avoids explicit formation of $\mathbf{\Psi}^\top \mathbf{\Psi}$ and therefore avoids the direct squaring of the conditioning associated with normal equations. This can improve attainable accuracy in finite precision, especially for moderately ill-conditioned collocation matrices or higher-degree fixed bases.

### 2.4. Extension to Inverse Problems

The same constrained algebraic structure extends naturally to inverse problems. We consider two settings: parameter identification, where an unknown scalar parameter enters the PDE operator, and source recovery, where the forcing term is unknown and must be inferred from measurements of the solution. In both cases, the boundary constraint block $\mathbf{\Phi}_b$ is independent of the inverse unknown. Hence the boundary conditions remain imposed through the same algebraic constraint $\mathbf{\Phi}_b \mathbf{C} = \mathbf{g}$ throughout the inversion.

**Parameter identification.** Suppose the differential operator depends on a scalar parameter $\theta$, for example a diffusion coefficient, so that $\mathbf{\Psi} = \mathbf{\Psi}(\theta)$. We are given $M$ sensor measurements of the solution,

$$d_k = u(\mathbf{x}_k) + \varepsilon_k, \qquad k = 1, \ldots, M, \tag{21}$$

where $\mathbf{x}_k \in \Omega$ are sensor locations and $\varepsilon_k$ denotes measurement noise. Let $\mathbf{S} \in \mathbb{R}^{M \times N}$ be the sensor evaluation matrix, with

$$S_{kj} = \phi_j(\mathbf{x}_k). \tag{22}$$

For a fixed value of $\theta$, we determine the coefficient vector $\mathbf{C}$ by minimizing both the PDE residual and the sensor-data misfit, while imposing the boundary conditions as hard constraints:

$$\min_{\mathbf{C}} \ \frac{1}{2}\|\mathbf{\Psi}(\theta)\mathbf{C} - \mathbf{f}\|_2^2 + \frac{1}{2}\|\mathbf{S}\mathbf{C} - \mathbf{d}\|_2^2 \quad \text{subject to} \quad \mathbf{\Phi}_b \mathbf{C} = \mathbf{g}. \tag{23}$$

The reduced KKT system for this fixed value of $\theta$ is

$$\begin{bmatrix} \mathbf{\Psi}(\theta)^\top \mathbf{\Psi}(\theta) + \mathbf{S}^\top \mathbf{S} & \mathbf{\Phi}_b^\top \\ \mathbf{\Phi}_b & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{C} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{\Psi}(\theta)^\top \mathbf{f} + \mathbf{S}^\top \mathbf{d} \\ \mathbf{g} \end{bmatrix}. \tag{24}$$

This system gives $\mathbf{C}(\theta)$ for the prescribed trial value of $\theta$. The unknown parameter is then recovered by minimizing the reduced objective

$$\widehat{\mathcal{J}}(\theta) = \frac{1}{2}\|\mathbf{\Psi}(\theta)\mathbf{C}(\theta) - \mathbf{f}\|_2^2 + \frac{1}{2}\|\mathbf{S}\mathbf{C}(\theta) - \mathbf{d}\|_2^2 \tag{25}$$

over a bounded interval. In the present work, since $\theta$ is scalar, this outer minimization is performed using Brent's method (Brent, 2013). Each function evaluation in the Brent search requires one constrained linear solve.

The augmented-system version is obtained by stacking the PDE and sensor residuals into a single least-squares residual. Define

$$\mathbf{A}_\theta = \begin{bmatrix} \mathbf{\Psi}(\theta) \\ \mathbf{S} \end{bmatrix}, \qquad \mathbf{b}_\theta = \begin{bmatrix} \mathbf{f} \\ \mathbf{d} \end{bmatrix}. \tag{26}$$

Then (23) is equivalently

$$\min_{\mathbf{C}} \ \frac{1}{2}\|\mathbf{A}_\theta \mathbf{C} - \mathbf{b}_\theta\|_2^2 \quad \text{subject to} \quad \mathbf{\Phi}_b \mathbf{C} = \mathbf{g}. \tag{27}$$

Introducing the residual variable $\mathbf{r}_\theta = \mathbf{A}_\theta \mathbf{C} - \mathbf{b}_\theta$ gives the augmented-system saddle-point form

$$\begin{bmatrix} \mathbf{I} & -\mathbf{A}_\theta & \mathbf{0} \\ \mathbf{A}_\theta^\top & \mathbf{0} & \mathbf{\Phi}_b^\top \\ \mathbf{0} & \mathbf{\Phi}_b & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r}_\theta \\ \mathbf{C} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{b}_\theta \\ \mathbf{0} \\ \mathbf{g} \end{bmatrix}. \tag{28}$$

Eliminating $\mathbf{r}_\theta$ from (28) recovers the reduced system (24). Thus both systems solve the same constrained inverse problem in exact arithmetic, but the augmented-system form avoids explicit formation of $\mathbf{A}_\theta^\top \mathbf{A}_\theta$.

**Source recovery.** We next consider the case where the forcing term $f$ is unknown. The goal is to infer a solution $\hat{u} = \boldsymbol{\phi}^\top \mathbf{C}$ from sensor measurements and then recover the source by applying the PDE operator to the learned representation. Thus, once $\mathbf{C}$ is obtained, the recovered source at the interior collocation points is evaluated as

$$\widehat{\mathbf{f}} = \boldsymbol{\Psi}\mathbf{C}. \tag{29}$$

Since source recovery involves differentiating the reconstructed solution, it is more ill-conditioned than direct solution recovery. We therefore use Tikhonov regularization and solve

$$\min_{\mathbf{C}} \frac{1}{2}\|\mathbf{S}\mathbf{C} - \mathbf{d}\|_2^2 + \frac{\beta}{2}\|\mathbf{C}\|_2^2 \quad \text{subject to} \quad \boldsymbol{\Phi}_b\mathbf{C} = \mathbf{g}, \tag{30}$$

where $\beta > 0$ is the Tikhonov parameter. The reduced KKT system is

$$\begin{bmatrix} \mathbf{S}^\top\mathbf{S} + \beta\mathbf{I} & \boldsymbol{\Phi}_b^\top \\ \boldsymbol{\Phi}_b & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{C} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{S}^\top\mathbf{d} \\ \mathbf{g} \end{bmatrix}. \tag{31}$$

Unlike parameter identification, source recovery does not require an outer search. For a chosen value of $\beta$, the coefficient vector $\mathbf{C}$ is obtained from a single constrained linear solve.

Tikhonov regularization is retained in the augmented-system formulation. It is written as an additional least-squares residual by defining

$$\mathbf{A}_\beta = \begin{bmatrix} \mathbf{S} \\ \sqrt{\beta}\mathbf{I} \end{bmatrix}, \qquad \mathbf{b}_\beta = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix}. \tag{32}$$

With this notation,

$$\frac{1}{2}\|\mathbf{S}\mathbf{C} - \mathbf{d}\|_2^2 + \frac{\beta}{2}\|\mathbf{C}\|_2^2 = \frac{1}{2}\|\mathbf{A}_\beta\mathbf{C} - \mathbf{b}_\beta\|_2^2. \tag{33}$$

Therefore, introducing $\mathbf{r}_\beta = \mathbf{A}_\beta\mathbf{C} - \mathbf{b}_\beta$ gives

$$\begin{bmatrix} \mathbf{I} & -\mathbf{A}_\beta & \mathbf{0} \\ \mathbf{A}_\beta^\top & \mathbf{0} & \boldsymbol{\Phi}_b^\top \\ \mathbf{0} & \boldsymbol{\Phi}_b & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r}_\beta \\ \mathbf{C} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{b}_\beta \\ \mathbf{0} \\ \mathbf{g} \end{bmatrix}. \tag{34}$$

Eliminating $\mathbf{r}_\beta$ from (34) recovers the reduced Tikhonov-regularized KKT system in (31). Thus the augmented-system formulation does not remove the Tikhonov term, but instead incorporates it through the additional residual $\sqrt{\beta}\mathbf{C}$.

Both inverse settings inherit the main advantage of the forward LM-PIELM formulation, whereby boundary conditions are not balanced against data misfit through an external penalty weight, but are imposed directly through the algebraic constraint $\boldsymbol{\Phi}_b\mathbf{C} = \mathbf{g}$. The augmented system form is used when avoiding explicit normal equation matrices is beneficial.

### 2.5. Solvability and Conditioning of the Discrete LM System

The LM-PIELM system is a discrete equality-constrained least-squares problem. The boundary conditions are enforced at the selected boundary collocation points through

$$\boldsymbol{\Phi}_b\mathbf{C} = \mathbf{g}. \tag{35}$$

Therefore, exact enforcement at these points requires the prescribed boundary data to be compatible with the chosen fixed-feature space, i.e.,

$$\mathbf{g} \in \mathcal{R}(\boldsymbol{\Phi}_b), \tag{36}$$

up to numerical precision. This condition does not require the rows of $\boldsymbol{\Phi}_b$ to be linearly independent. Hence the boundary may be oversampled, and some boundary equations may be redundant. In that case, the Lagrange multipliers need not be unique, but the primal coefficient vector $\mathbf{C}$ can still be well determined.

A sufficient condition for uniqueness of the primal coefficient vector is

$$\mathcal{N}(\boldsymbol{\Psi}) \cap \mathcal{N}(\boldsymbol{\Phi}_b) = \{\mathbf{0}\}. \tag{37}$$

This means that no nonzero coefficient perturbation can leave both the interior PDE residual and the enforced boundary constraints unchanged. In other words, after the boundary constraints are imposed, the remaining degrees of freedom must be identifiable from the PDE residual. For the inverse formulations, $\boldsymbol{\Psi}$ is replaced by the corresponding least-squares matrix, such as $\mathbf{A}_\theta$ or $\mathbf{A}_\beta$.

In the numerical implementation, the assembled saddle-point system is solved using a least-squares solver based on complete orthogonal factorization (via the LAPACK gelsy driver), which also handles possible rank deficiency or redundant equations by returning a minimum-norm solution. Boundary feasibility is monitored directly through $\|\boldsymbol{\Phi}_b\mathbf{C} - \mathbf{g}\|_\infty$.

## 3. Numerical Experiments

### 3.1. Experimental Setup

We validate LM-PIELM on three forward and three inverse benchmarks of increasing complexity. An additional forward experiment (1D unsteady advection-diffusion in space-time) is reported in Appendix A, and two additional inverse

experiments (conductivity identification in 1D heat conduction and source recovery for the 2D Helmholtz equation) are reported in Appendix B.

Each problem is solved using both the reduced KKT form (12) and the augmented saddle-point form (20), referred to as **LM-PIELM** and **LM-PIELM (Aug.)** respectively throughout all tables and discussion. All figures show the LM-PIELM (Aug.) solution unless noted otherwise.

In all cases the solution is approximated with a single-hidden-layer network whose fixed feature layer consists of tensor-product Bernstein polynomials. The linear-in-parameters structure that underlies LM-PIELM holds for any fixed basis. Bernstein polynomials provide a deterministic, well-conditioned alternative, and the framework applies without modification to random-feature ELMs. The assembled saddle-point system is solved via a least-squares routine (`scipy.linalg.lstsq`). All computations run on a single CPU core (Intel i5-13450HX, 2.40 GHz, 16 GB RAM) with no GPU acceleration. For inverse problems, sensor locations are sampled uniformly at random within $\Omega$, and measurements are corrupted by additive Gaussian noise $\varepsilon_k \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = \eta \cdot \|u_{\text{exact}}\|_\infty$, where $\eta$ denotes the noise level (e.g. $\eta = 0.01$ for 1% noise).

### 3.2. Forward Problems

We first test the method on three forward problems. Where available, we compare against X-TFC (De Florio et al., 2024; Schiassi et al., 2021) and standard PIELM. We report the the solution error $\|u_{\text{ex}} - \hat{u}\|_\infty$ on a dense test grid and the discrete boundary residual $\|\mathbf{\Phi}_b\mathbf{C} - \mathbf{g}\|_\infty$. For X-TFC, boundary constraints are embedded analytically and are therefore reported as *Exact*.

#### 3.2.1. Steady 1D Advection-Diffusion

We consider $u_x - \nu\, u_{xx} = f(x)$ on $(0, 1)$ with Dirichlet conditions $u(0) = 0$, $u(1) = 1$. The exact solution is $u_{\text{exact}}(x) = (e^{x/\nu} - 1)/(e^{1/\nu} - 1)$, which develops an increasingly sharp boundary layer near $x = 1$ as $\nu$ decreases. We test the advection-dominated regimes $\nu = 0.01$, $0.001$, and $0.0001$. For X-TFC and standard PIELM, we use our in-house implementations with the basis and collocation configurations reported by De Florio et al. (2024). For LM-PIELM and LM-PIELM (Aug.), the same configuration is used for both algebraic forms at each value of $\nu$: $(N, N_c, N_b) = (100, 1500, 2)$ for $\nu = 0.01$, $(1200, 2500, 2)$ for $\nu = 0.001$, and $(3500, 7000, 2)$ for $\nu = 0.0001$. In all cases, the interior collocation points are sampled uniformly in $(0, 1)$, and accuracy is evaluated on a uniform test grid of 10,000 points.

Table 1 shows that both LM-PIELM forms perform very well for $\nu = 0.01$, with solution errors of order $10^{-14}$ and boundary residuals at roundoff level. As the problem becomes more advection dominated, the advantage of the augmented form becomes clear. At $\nu = 0.001$, LM-PIELM (Aug.) improves the solution error from $4.04 \times 10^{-11}$ to $9.34 \times 10^{-13}$ compared with the reduced KKT form, while both methods retain near-machine-precision boundary enforcement. At the most severe case, $\nu = 0.0001$, X-TFC and standard PIELM also lose accuracy. The reduced LM-PIELM system improves over standard PIELM but still gives a solution error of $7.17 \times 10^{-2}$ and boundary residual $1.78 \times 10^{-5}$. In contrast, LM-PIELM (Aug.) gives $1.61 \times 10^{-4}$ solution error and keeps the boundary residual at $9.31 \times 10^{-14}$, reflecting the conditioning benefit of avoiding the Gram matrix $\mathbf{\Psi}^\top\mathbf{\Psi}$. Figure 1 shows the LM-PIELM (Aug.) prediction for the most advection-dominated case, $\nu = 0.0001$.

The augmented form is more expensive, with wall-clock times increasing from 0.22 s to 70 s as $\nu$ decreases from 0.01 to 0.0001, compared with 0.04-6.24 s for the reduced KKT form. This cost is due to the larger system dimension $N_c + N + N_b$, but it gives the largest stability and accuracy gain in the most advection-dominated case.

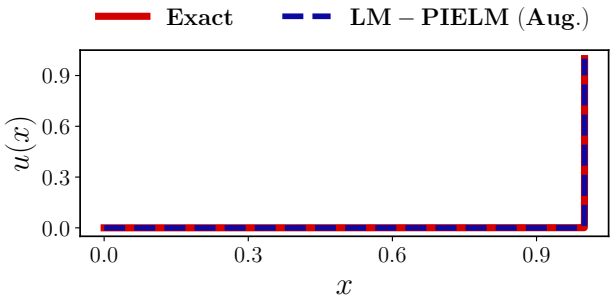
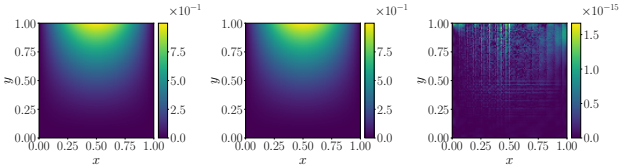



*Figure 1.* Steady 1D advection-diffusion problem at $\nu = 0.0001$, showing the exact solution and the LM-PIELM (Aug.) prediction.

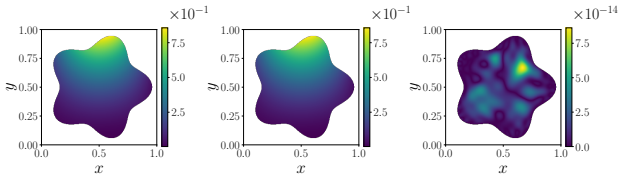
*Table 1.* Steady 1D advection-diffusion problem, reporting solution errors and boundary constraint residuals for different diffusion coefficients.

| $\nu$ | Method | $\|u_{\text{ex}} - \hat{u}\|_\infty$ | $\|\Phi_b\mathbf{C} - \mathbf{g}\|_\infty$ |
|---|---|---|---|
| 0.01 | X-TFC | $1.73\times10^{-12}$ | Exact |
| | PIELM | $2.18\times10^{-8}$ | $2.18\times10^{-8}$ |
| | LM-PIELM | $1.87\times10^{-14}$ | $2.22\times10^{-16}$ |
| | LM-PIELM (Aug.) | $\mathbf{1.27\times10^{-14}}$ | $\mathbf{2.21\times10^{-16}}$ |
| 0.001 | X-TFC | $2.83\times10^{-12}$ | Exact |
| | PIELM | $4.89\times10^{-5}$ | $4.89\times10^{-5}$ |
| | LM-PIELM | $4.04\times10^{-11}$ | $6.66\times10^{-16}$ |
| | LM-PIELM (Aug.) | $\mathbf{9.34\times10^{-13}}$ | $\mathbf{2.22\times10^{-16}}$ |
| 0.0001 | X-TFC | $2.89\times10^{-3}$ | Exact |
| | PIELM | $5\times10^{-1}$ | $4.89\times10^{-1}$ |
| | LM-PIELM | $7.17\times10^{-2}$ | $1.78\times10^{-5}$ |
| | LM-PIELM (Aug.) | $\mathbf{1.61\times10^{-4}}$ | $\mathbf{9.31\times10^{-14}}$ |




(a) Exact (b) LM – PIELM (Aug.) (c) Absolute Error

*Figure 2.* 2D Poisson equation with mixed boundary conditions: exact solution, LM-PIELM (Aug.) prediction, and absolute error.




(a) Exact (b) LM – PIELM (c) Absolute Error

*Figure 3.* 2D Poisson equation on a star-shaped domain: exact solution, LM-PIELM (Aug.) prediction, and absolute error.

#### 3.2.2. 2D Poisson Equation with Mixed Boundary Conditions

We solve $u_{xx} + u_{yy} = f(x, y)$ on $(0, 1)^2$, with homogeneous Dirichlet conditions on three sides and the Neumann condition $\partial u/\partial y(x, 1) = 2\sin(\pi x)$ on the top boundary. The exact solution is $u_{\text{ex}}(x, y) = y^2 \sin(\pi x)$, giving $f = (2 - \pi^2 y^2)\sin(\pi x)$. We use $N = 289$ tensor-product Bernstein basis functions, $N_c = 2304$ uniformly sampled interior collocation points, and $N_b = 196$ boundary points. Accuracy is evaluated on a $100 \times 100$ test grid.

The mixed-BC rows in Table 2 show that LM-PIELM (Aug.) reduces both the solution and boundary errors to near double-precision roundoff. Fig. 2 shows no visible error accumulation near the Neumann boundary, indicating that different boundary operators are enforced together in the same Lagrange block without separate penalty weights. The wall-clock times for LM-PIELM and LM-PIELM (Aug.) are 1.31 s and 2.26 s, respectively.

#### 3.2.3. 2D Poisson Equation on a Star-Shaped Domain

We next solve the same Poisson equation on a star-shaped domain $\Omega \subset (0, 1)^2$ with boundary $r(\theta) = R(1+\alpha\cos k\theta)$, centred at $(0.5, 0.5)$, where $R = 0.40$, $\alpha = 0.15$, and $k = 5$. Dirichlet data $u = u_{\text{ex}}$ are imposed on the full boundary, with $u_{\text{ex}} = y^2 \sin(\pi x)$ and $f = (2 - \pi^2 y^2)\sin(\pi x)$. We use the same $N = 289$ basis functions, $N_c = 3168$ interior points rejection-sampled from an $80 \times 80$ grid, and $N_b = 400$ boundary points sampled parametrically on $\partial\Omega$. Accuracy is evaluated on a $200 \times 200$ test grid inside the physical domain. This benchmark tests whether the constraint assembly is independent of the geometry. The algebraic formulation is unchanged from the square domain. No X-TFC result is reported because TFC constrained expressions are geometry dependent and are not readily available for this boundary. The star-domain rows in Table 2 show that LM-PIELM (Aug.) achieves $10^{-14}$-level solution accuracy with near-machine-precision boundary residuals. Fig. 3 confirms that the error remains small without visible artefacts near the curved boundary. The wall-clock times for LM-PIELM and LM-PIELM (Aug.) are 1.81 s and 7.84 s, respectively.

*Table 2.* Forward 2D Poisson benchmarks, reporting solution errors and boundary constraint residuals for the mixed-boundary and star-shaped domain cases.

| Case | Method | $\|u_{\text{ex}} - \hat{u}\|_\infty$ | $\|\Phi_b \mathbf{C} - \mathbf{g}\|_\infty$ |
|---|---|---|---|
| Mixed BC | X-TFC | $7.58 \times 10^{-12}$ | Exact |
| | PIELM | $1.78 \times 10^{-10}$ | $1.68 \times 10^{-9}$ |
| | LM-PIELM | $2.79 \times 10^{-12}$ | $5.61 \times 10^{-13}$ |
| | LM-PIELM (Aug.) | $\mathbf{1.67 \times 10^{-15}}$ | $\mathbf{6.51 \times 10^{-15}}$ |
| Star domain | PIELM | $1.49 \times 10^{-12}$ | $1.49 \times 10^{-12}$ |
| | LM-PIELM | $6.47 \times 10^{-11}$ | $2.92 \times 10^{-11}$ |
| | LM-PIELM (Aug.) | $\mathbf{4.50 \times 10^{-14}}$ | $\mathbf{1.33 \times 10^{-15}}$ |

### 3.3. Inverse Problems

We evaluate LM-PIELM on two inverse benchmarks: one scalar parameter-identification problem and one source-recovery problem on an irregular domain. In both cases, the Lagrange constraint block $\mathbf{\Phi}_b$ is independent of the inverse unknown and is preserved throughout the inversion. For parameter identification, we report the relative parameter error $|\theta_{\text{rec}} - \theta_{\text{true}}|/|\theta_{\text{true}}|$ and the boundary constraint residual $\|\mathbf{\Phi}_b \mathbf{C} - \mathbf{g}\|_\infty$. For source recovery, the primary metric is the relative source error $\|f_{\text{ex}} - \hat{f}\|_2/\|f_{\text{ex}}\|_2$, reported in the $L^2$ norm because applying the differential operator to the recovered solution amplifies measurement errors. Additional inverse benchmarks for heat-conduction parameter recovery and Helmholtz source recovery are reported in Appendix B.

#### 3.3.1. Diffusivity Identification in Steady 1D Advection-Diffusion

We recover the unknown diffusion coefficient $\nu$ in $u_x - \nu u_{xx} = f(x)$ from sparse noisy measurements of $u$, with $\nu_{\text{true}} = 0.1$. This inverse benchmark uses $N = 25$ Bernstein basis functions, $N_c = 1000$ interior collocation points. The operator is decomposed as

$$\mathbf{\Psi}(\nu) = \mathbf{\Psi}_{\text{adv}} - \nu \mathbf{\Psi}_{\text{diff}},$$

and $\nu$ is recovered using Brent's method (**?**) over $\nu \in [10^{-3}, 0.9]$. Sensors are placed uniformly in $(0, 1)$. Each Brent iteration requires one constrained linear solve, and the total wall-clock time remains below 0.1 s for LM-PIELM and below 7.6 s for LM-PIELM (Aug.) for all tested configurations.

*Table 3.* Inverse diffusivity recovery for the steady 1D advection-diffusion problem with $\nu_{\text{true}} = 0.1$, reporting relative parameter errors and boundary constraint residuals for different sensor counts and noise levels.

| | | LM-PIELM | | LM-PIELM (AUG.) | |
|---|---|---|---|---|---|
| $M$ | $\eta$ | $\frac{\lvert\nu_{\text{rec}}-\nu_{\text{true}}\rvert}{\lvert\nu_{\text{true}}\rvert}$ | $\lVert\Phi_b\mathbf{C}-\mathbf{g}\rVert_\infty$ | $\frac{\lvert\nu_{\text{rec}}-\nu_{\text{true}}\rvert}{\lvert\nu_{\text{true}}\rvert}$ | $\lVert\Phi_b\mathbf{C}-\mathbf{g}\rVert_\infty$ |
| 5 | 0% | $7.60\times10^{-10}$ | $2.22\times10^{-16}$ | $7.60\times10^{-10}$ | $2.22\times10^{-16}$ |
| 100 | 0% | $4.51\times10^{-9}$ | $4.44\times10^{-16}$ | $4.51\times10^{-9}$ | $6.66\times10^{-16}$ |
| 20 | 1% | $6.10\times10^{-3}$ | $5.55\times10^{-16}$ | $6.10\times10^{-3}$ | $4.44\times10^{-16}$ |
| 200 | 1% | $6.62\times10^{-4}$ | $4.44\times10^{-16}$ | $6.62\times10^{-3}$ | $4.44\times10^{-16}$ |
| 50 | 5% | $2.31\times10^{-2}$ | $4.44\times10^{-16}$ | $2.31\times10^{-2}$ | $2.22\times10^{-16}$ |
| 100 | 5% | $9.26\times10^{-5}$ | $2.22\times10^{-16}$ | $9.26\times10^{-3}$ | $1.08\times10^{-16}$ |

*Table 4.* Source recovery for the 2D Poisson equation on the star-shaped domain, reporting relative source errors and boundary constraint residuals for different sensor counts and noise levels.

| | | LM-PIELM | | LM-PIELM (AUG.) | |
|---|---|---|---|---|---|
| $M$ | $\eta$ | $\frac{\lVert f_{\text{ex}}-\hat{f}\rVert_2}{\lVert f_{\text{ex}}\rVert_2}$ | $\lVert\Phi_b\mathbf{C}-\mathbf{g}\rVert_\infty$ | $\frac{\lVert f_{\text{ex}}-\hat{f}\rVert_2}{\lVert f_{\text{ex}}\rVert_2}$ | $\lVert\Phi_b\mathbf{C}-\mathbf{g}\rVert_\infty$ |
| 50 | 0% | $2.89\times10^{-3}$ | $1.33\times10^{-15}$ | $5.23\times10^{-4}$ | $9.99\times10^{-16}$ |
| 500 | 0% | $9.34\times10^{-5}$ | $9.99\times10^{-16}$ | $9.93\times10^{-6}$ | $9.99\times10^{-16}$ |
| 200 | 1% | $1.01\times10^{0}$ | $1.92\times10^{-11}$ | $1.01\times10^{0}$ | $1.92\times10^{-11}$ |
| 1000 | 1% | $6.36\times10^{-1}$ | $4.06\times10^{-11}$ | $6.35\times10^{-1}$ | $7.04\times10^{-12}$ |
| 200 | 5% | $8.43\times10^{-1}$ | $1.60\times10^{-11}$ | $8.44\times10^{-1}$ | $1.57\times10^{-11}$ |
| 500 | 5% | $5.87\times10^{-1}$ | $2.16\times10^{-11}$ | $5.87\times10^{-1}$ | $2.01\times10^{-11}$ |

Table 3 reports representative sensor/noise configurations. In the noiseless setting, both algebraic forms recover $\nu$ to relative errors of order $10^{-9}$. Under noisy data, the parameter error increases with the noise level, as expected. The main point is the boundary column. The constraint residual remains at $10^{-15}$-$10^{-16}$ for both LM-PIELM and LM-PIELM (Aug.), showing that the outer parameter search does not compromise boundary admissibility.

#### 3.3.2. Source Recovery in the 2D Poisson Equation on a Star-Shaped Domain

We next recover the source term in the Poisson equation on the star-shaped domain used in Section 3.2.3. The exact solution is $u_{\text{ex}}(x, y) = y^2 \sin(\pi x)$, giving $f_{\text{ex}}(x, y) = (2 - \pi^2 y^2) \sin(\pi x)$, with Dirichlet data $g = u_{\text{ex}}|_{\partial\Omega}$. The same constrained algebraic system is used without structural modification. The configuration is kept the same for both LM-PIELM and LM-PIELM (Aug.). We use $13 \times 13$ tensor-product Bernstein basis functions ($N = 169$ total basis functions), $N_c = 2420$ interior collocation points obtained by rejection sampling from a $70 \times 70$ grid, and $N_b = 400$ boundary points sampled parametrically on $\partial\Omega$. The test set contains 22 122 interior points and 2000 boundary points.

This case is particularly demanding, as it involves noisy measurements, second-derivative recovery, and an irregular geometry. For LM-PIELM, we use $\beta = 10^{-10}$ in the noiseless case and $\beta = 10^{-2}$ under noise. For LM-PIELM (Aug.), we use $\beta = 10^{-12}$ in the noiseless case and $\beta = 10^{-2}$ under noise. The wall-clock time per solve ranges from 0.05-0.16 s for LM-PIELM and 0.22-1.2 s for LM-PIELM (Aug.).

Table 4 shows that, in the noiseless setting, LM-PIELM (Aug.) gives lower source errors than the reduced KKT form. Under noise, the source error increases because source recovery requires applying the differential operator to a solution inferred from noisy measurements. However, the boundary residual remains small across all configurations. This shows that the degradation in the recovered source is confined to the interior reconstruction, while boundary admissibility is still enforced by the same algebraic Lagrange block. Thus, even on the irregular geometry, the boundary enforcement remains hard and does not require any change to the formulation.

## 4. Conclusion

We have presented LM-PIELM, a Lagrange-multiplier formulation of physics-informed extreme learning machines in which boundary conditions are imposed as algebraic equality constraints rather than soft penalty terms. The method preserves the fixed-feature, linear-in-parameters structure of PIELM, so the constrained PDE problem reduces to a linear saddle-point system. Since the constraint block is assembled directly from boundary operator evaluations, Dirichlet, Neumann, and mixed boundary conditions can be treated within the same algebraic framework, without penalty tuning, iterative dual updates, or problem-specific analytical constrained constructions.

We also used an augmented-system implementation of the same constrained least-squares problem. This form avoids explicit formation of the Gram matrix $\mathbf{\Psi}^\top\mathbf{\Psi}$ and improves robustness in ill-conditioned regimes, while remaining algebraically equivalent to the reduced KKT form in exact arithmetic. Across the forward benchmarks, LM-PIELM (Aug.) improves robustness and solution accuracy in ill-conditioned regimes, while preserving small boundary residuals. This confirms the practical benefit of avoiding explicit normal-equation Gram matrices. The method matches existing hard-constrained approaches on standard geometries and extends directly to the star-shaped domain, showing that the constraint enforcement is not tied to a specific domain geometry.

For inverse problems, the same constraint block is retained throughout the recovery procedure, so boundary enforcement is not re-balanced against noisy sensor data or regularization terms. Although parameter and source recovery degrade under noise, the boundary residuals remain small across the tested sensor counts, noise levels, and geometries. Additional forward and inverse experiments in Ap-

pendices A and B show the same behaviour across other operators and problem settings. Future work will extend the framework to evolving-boundary and interface problems, as well as to time-dependent and multiphysics settings where hard enforcement of boundary, interface, or coupling conditions is critical.

## Impact Statement

This work is methodological and aims to improve the reliability of physics-informed solvers by enforcing boundary conditions as hard constraints. Such developments can support computational science and engineering applications where repeated PDE solves are required, for example in design, inverse modelling, and simulation-based analysis. The expected impact is therefore mainly through more accurate and stable numerical tools. We do not foresee any direct negative societal impact, apart from the usual risk that computational predictions may be used without sufficient validation for the problem of interest.

## References


Benzi, M., Golub, G. H., and Liesen, J. Numerical solution of saddle point problems. *Acta numerica*, 14:1–137, 2005.

Brent, R. P. *Algorithms for minimization without derivatives*. Courier Corporation, 2013.

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022.

De Florio, M., Schiassi, E., Calabrò, F., and Furfaro, R. Physics-informed neural networks for 2nd order odes with sharp gradients. *Journal of Computational and Applied Mathematics*, 436:115396, 2024.

Dong, S. and Ni, N. A method for representing periodic functions and enforcing exactly periodic boundary conditions with deep neural networks. *Journal of Computational Physics*, 435:110242, 2021.

Dwivedi, V. and Srinivasan, B. Physics informed extreme learning machine (pielm)–a rapid method for the numerical solution of partial differential equations. *Neurocomputing*, 391:96–118, 2020.

E, W. and Yu, B. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6 (1):1–12, 2018.

Golub, G. H. and Van Loan, C. F. *Matrix computations*. JHU press, 2013.

Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. Extreme learning machine: theory and applications. *Neurocomputing*, 70 (1–3):489–501, 2006.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Kharazmi, E., Zhang, Z., and Karniadakis, G. E. hp-vpinns: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 374:113547, 2021.

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R. M., and Mahoney, M. W. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 26548–26560, 2021.

Lagaris, I. E., Likas, A., and Fotiadis, D. I. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5): 987–1000, 1998.

Liu, H., Xing, B., Wang, Z., and Li, L. Legendre neural network method for several classes of singularly perturbed differential equations based on mapping and piecewise optimization technology. *Neural Processing Letters*, 51 (3):2891–2913, 2020.

Liu, M., Hou, M., Wang, J., and Cheng, Y. Solving two-dimensional linear partial differential equations based on chebyshev neural network with extreme learning machine algorithm. *Engineering Computations*, 38(2):874–894, 2021.

Mishra, R., Krishnamurthi, G., Srinivasan, B., Natarajan, S., et al. Eig-pielm: an efficient mesh-free method for eigenvalue problems using physics-informed extreme learning machines. *Computer Methods in Applied Mechanics and Engineering*, 451:118674, 2026.

Mortari, D. The theory of connections: Connecting points. *Mathematics*, 5(4):57, 2017.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. 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.

Schiassi, E., Florio, R., D'Ambrosio, A., Mortari, D., and Furfaro, R. Extreme theory of functional connections: A fast physics-informed neural network method for solving ordinary and partial differential equations. *Neurocomputing*, 457:334–356, 2021.

Son, H., Cho, S. W., and Hwang, H. J. Enhanced physics-informed neural networks with augmented lagrangian relaxation method (al-pinns). *Neurocomputing*, 548: 126424, 2023.

Sukumar, N. and Srivastava, A. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods in Applied Mechanics and Engineering*, 389:114333, 2022.

Sun, H., Hou, M., Yang, Y., Zhang, T., Weng, F., and Han, F. Solving partial differential equation based on bernstein neural network and extreme learning machine algorithm. *Neural Processing Letters*, 50(2):1153–1172, 2019.

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

Xiang, Z., Peng, W., Liu, X., and Yao, W. Self-adaptive loss balanced physics-informed neural networks. *Neurocomputing*, 496:11–34, 2022.

Yang, Y., Hou, M., and Luo, J. A novel improved extreme learning machine algorithm in solving ordinary differential equations by legendre neural network methods. *Advances in Difference Equations*, 2018(1):469, 2018.

Yang, Y., Hou, M., Sun, H., Zhang, T., Weng, F., and Luo, J. Neural network algorithm based on legendre improved extreme learning machine for solving elliptic partial differential equations. *Soft Computing*, 24(2):1083–1096, 2020.

# A. Supplementary Forward Experiments

This appendix reports an additional forward benchmark that complements Section 3.2. The notation, error metrics, and method names are kept the same as in the main text.

## A.1. 1D Unsteady Advection-Diffusion

We consider a one-dimensional unsteady advection-diffusion problem in a space-time setting,

$$u_t(x,t) + c\,u_x(x,t) - \kappa\,u_{xx}(x,t) = f(x,t), \qquad (x,t) \in [0,1] \times [0,1], \tag{38}$$

with advection speed $c = 1.0$ and diffusion coefficient $\kappa = 0.05$. The exact solution is

$$u_{\text{ex}}(x,t) = \sin\big(2\pi(x - ct)\big)\, e^{-\alpha t}, \qquad \alpha = 0.5, \tag{39}$$

which represents a travelling sinusoidal profile with exponential decay. The forcing term is obtained by substituting the exact solution into the PDE,

$$f(x,t) = \big(-\alpha + \kappa(2\pi)^2\big)\sin\big(2\pi(x - ct)\big)e^{-\alpha t}. \tag{40}$$

Dirichlet conditions are imposed at $x = 0$ and $x = 1$ for all $t \in [0,1]$, and the initial condition is prescribed at $t = 0$. In the space-time formulation, the two boundary traces and the initial condition are all included as rows of the same Lagrange constraint block.

We use tensor-product Bernstein polynomials with 22 basis functions per axis, giving $N = 484$ basis functions. The PDE residual is collocated on a uniform $70 \times 70$ space-time grid, giving 4692 interior collocation points after removing the constrained points. The boundary and initial-condition constraints are imposed at 208 points sampled from $\{x = 0\} \cup \{x = 1\} \cup \{t = 0\}$. The assembly and solve time is 0.7601 s for LM-PIELM and 10.02 s for LM-PIELM (Aug.) on the same single-CPU setup used in the main text. Table 5 reports the solution error and boundary constraint residual at three time

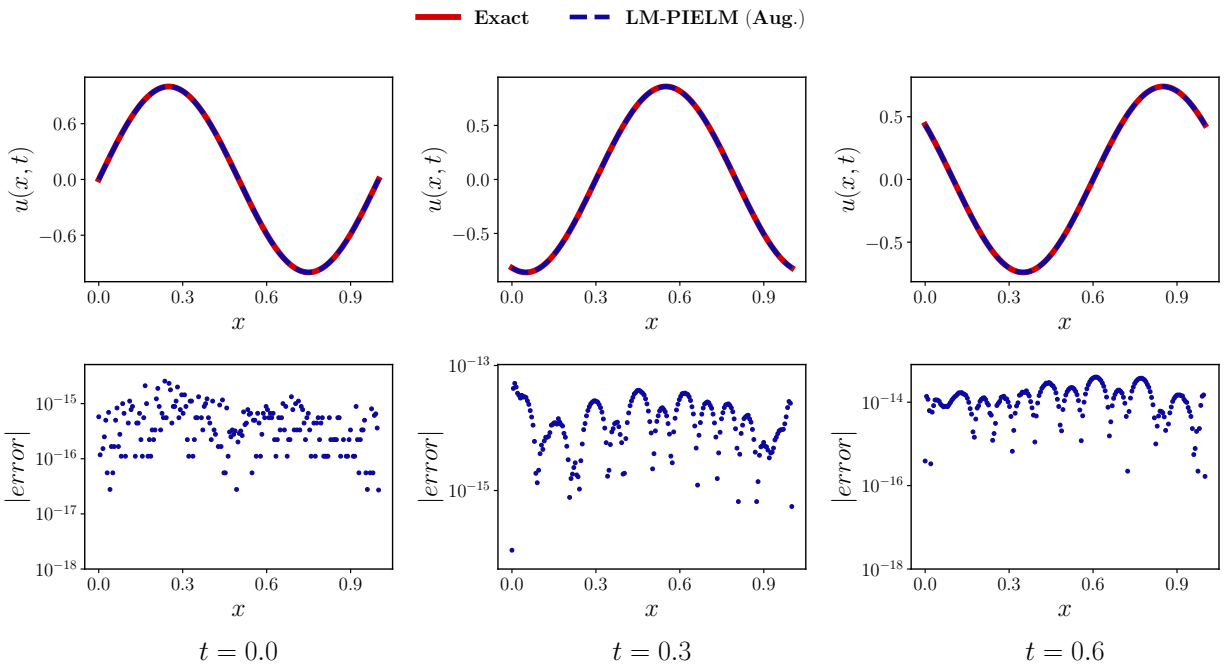



*Figure 4.* Unsteady 1D advection-diffusion problem in space-time, showing the exact solution, LM-PIELM (Aug.) prediction, and pointwise absolute error at selected time instances.

instances. LM-PIELM already gives small errors, but the augmented form is consistently more accurate. At $t = 0$, the error is $2.78 \times 10^{-15}$, reflecting the algebraic enforcement of the initial condition. At later times, the error remains at $10^{-14}$, while the boundary residual stays at machine precision. Figure 4 shows that the predicted profiles are visually indistinguishable from the exact solution, and that the pointwise errors remain small throughout the spatial interval.

*Table 5.* Unsteady 1D advection-diffusion problem in space-time, reporting solution errors for LM-PIELM and LM-PIELM (Aug.) at selected time instances.

| $t$ | METHOD | $\|u_{\text{ex}} - \hat{u}\|_\infty$ | $\|\Phi_b \mathbf{C} - \mathbf{g}\|_\infty$ |
|---|---|---|---|
| 0.0 | LM-PIELM | $6.00\times10^{-14}$ | $5.08\times10^{-14}$ |
| | LM-PIELM (AUG.) | $\mathbf{2.78\times10^{-15}}$ | $\mathbf{5.88\times10^{-16}}$ |
| 0.3 | LM-PIELM | $1.26\times10^{-10}$ | $4.88\times10^{-15}$ |
| | LM-PIELM (AUG.) | $\mathbf{5.20\times10^{-14}}$ | $\mathbf{4.44\times10^{-16}}$ |
| 0.6 | LM-PIELM | $1.05\times10^{-10}$ | $1.61\times10^{-15}$ |
| | LM-PIELM (AUG.) | $\mathbf{4.02\times10^{-14}}$ | $\mathbf{2.77\times10^{-16}}$ |

*Table 6.* Inverse conductivity recovery for the steady 1D heat conduction problem with $\kappa_{\text{true}} = 1.0$, reporting relative conductivity errors, solution errors, and boundary constraint residuals.

| | | LM-PIELM | | | LM-PIELM (AUG.) | | |
|---|---|---|---|---|---|---|---|
| $M$ | $\eta$ | $\frac{\lvert\kappa_{\text{rec}} - \kappa_{\text{true}}\rvert}{\lvert\kappa_{\text{true}}\rvert}$ | $\|T_{\text{ex}} - \hat{T}\|_\infty$ | $\|\Phi_b \mathbf{C} - \mathbf{g}\|_\infty$ | $\frac{\lvert\kappa_{\text{rec}} - \kappa_{\text{true}}\rvert}{\lvert\kappa_{\text{true}}\rvert}$ | $\|T_{\text{ex}} - \hat{T}\|_\infty$ | $\|\Phi_b \mathbf{C} - \mathbf{g}\|_\infty$ |
| 5 | 0% | $1.39\times10^{-9}$ | $1.74\times10^{-10}$ | $5.83\times10^{-17}$ | $1.39\times10^{-9}$ | $1.74\times10^{-10}$ | $4.04\times10^{-16}$ |
| 10 | 0% | $1.39\times10^{-9}$ | $1.74\times10^{-10}$ | $5.03\times10^{-17}$ | $1.39\times10^{-9}$ | $1.75\times10^{-10}$ | $2.19\times10^{-16}$ |
| 20 | 0% | $1.39\times10^{-9}$ | $1.74\times10^{-10}$ | $7.13\times10^{-17}$ | $1.40\times10^{-9}$ | $1.75\times10^{-10}$ | $2.89\times10^{-17}$ |
| 50 | 0% | $1.40\times10^{-9}$ | $1.75\times10^{-10}$ | $7.26\times10^{-17}$ | $1.40\times10^{-9}$ | $1.75\times10^{-10}$ | $1.62\times10^{-16}$ |
| 100 | 0% | $1.41\times10^{-9}$ | $1.76\times10^{-10}$ | $3.64\times10^{-17}$ | $1.41\times10^{-9}$ | $1.76\times10^{-10}$ | $4.64\times10^{-17}$ |
| 200 | 0% | $1.42\times10^{-9}$ | $1.77\times10^{-10}$ | $3.29\times10^{-17}$ | $1.42\times10^{-9}$ | $1.78\times10^{-10}$ | $2.98\times10^{-16}$ |
| 5 | 1% | $8.36\times10^{-4}$ | $1.04\times10^{-4}$ | $4.33\times10^{-17}$ | $4.41\times10^{-3}$ | $5.54\times10^{-4}$ | $1.58\times10^{-16}$ |
| 10 | 1% | $2.16\times10^{-3}$ | $2.71\times10^{-4}$ | $9.88\times10^{-17}$ | $1.45\times10^{-3}$ | $1.81\times10^{-4}$ | $1.63\times10^{-16}$ |
| 20 | 1% | $8.47\times10^{-4}$ | $1.06\times10^{-4}$ | $1.42\times10^{-16}$ | $2.58\times10^{-3}$ | $3.22\times10^{-4}$ | $5.20\times10^{-16}$ |
| 50 | 1% | $3.10\times10^{-3}$ | $3.86\times10^{-4}$ | $1.39\times10^{-17}$ | $3.78\times10^{-4}$ | $4.72\times10^{-5}$ | $3.37\times10^{-16}$ |
| 100 | 1% | $8.10\times10^{-4}$ | $1.01\times10^{-4}$ | $2.19\times10^{-17}$ | $1.55\times10^{-4}$ | $1.93\times10^{-5}$ | $7.63\times10^{-17}$ |
| 200 | 1% | $7.39\times10^{-4}$ | $9.24\times10^{-5}$ | $8.67\times10^{-17}$ | $1.20\times10^{-3}$ | $1.49\times10^{-4}$ | $5.19\times10^{-16}$ |
| 5 | 5% | $6.09\times10^{-3}$ | $7.66\times10^{-4}$ | $6.97\times10^{-17}$ | $2.23\times10^{-2}$ | $2.85\times10^{-3}$ | $2.74\times10^{-16}$ |
| 10 | 5% | $1.37\times10^{-2}$ | $1.74\times10^{-3}$ | $3.64\times10^{-17}$ | $5.52\times10^{-3}$ | $6.86\times10^{-4}$ | $1.54\times10^{-16}$ |
| 20 | 5% | $2.44\times10^{-3}$ | $3.06\times10^{-4}$ | $1.57\times10^{-17}$ | $1.48\times10^{-2}$ | $1.82\times10^{-3}$ | $2.36\times10^{-16}$ |
| 50 | 5% | $1.70\times10^{-2}$ | $2.16\times10^{-3}$ | $3.12\times10^{-17}$ | $4.45\times10^{-3}$ | $5.59\times10^{-4}$ | $4.49\times10^{-16}$ |
| 100 | 5% | $5.01\times10^{-3}$ | $6.23\times10^{-4}$ | $6.53\times10^{-17}$ | $1.58\times10^{-3}$ | $1.98\times10^{-4}$ | $3.57\times10^{-16}$ |
| 200 | 5% | $4.48\times10^{-3}$ | $5.58\times10^{-4}$ | $5.67\times10^{-17}$ | $8.01\times10^{-4}$ | $1.00\times10^{-4}$ | $2.24\times10^{-16}$ |

## B. Supplementary Inverse Experiments

This appendix reports two additional inverse benchmarks that complement Section 3.3. The heat-conduction case extends the 1D parameter-identification results, while the Helmholtz case extends the source-recovery study to a reaction-diffusion operator. The notation, method names, and error metrics follow the main text.

### B.1. Conductivity Identification in Steady 1D Heat Conduction

We recover the unknown conductivity $\kappa$ in $-\kappa T_{xx} = 1$ from sparse noisy measurements of $T(x)$, with $\kappa_{\text{true}} = 1.0$. The exact solution is $T(x) = x(1-x)/(2\kappa)$. We use $N = 25$ Bernstein polynomials (degree 24), $N_c = 1000$ interior collocation points, and $N_b = 2$ boundary points. The operator is $\mathbf{\Psi}(\kappa) = -\kappa\mathbf{\Phi}_{xx}$, so the inverse problem reduces to a scalar Brent search over $\kappa \in [0.1, 20]$. Sensors are placed uniformly in $(0, 1)$, and accuracy is evaluated on 2000 test points.

Table 6 reports the full sweep over $M \in \{5, 10, 20, 50, 100, 200\}$ and $\eta \in \{0\%, 1\%, 5\%\}$. In the noiseless setting, both forms recover $\kappa$ with relative errors of order $10^{-9}$, and the temperature error stays below $10^{-9}$. Under $1\%$ and $5\%$ noise, the parameter and solution errors increase as expected, with variations mainly governed by the noise realisation. Since $\kappa$ enters only as a scalar multiplier of the second-derivative block, the parameter and solution errors follow nearly the same trend.

The main observation is unchanged from the inverse examples in the main paper. The boundary residual remains at $10^{-16}$-$10^{-17}$ for both LM-PIELM and LM-PIELM (Aug.) across all sensor counts and noise levels. Thus the recovered temperature field remains boundary-admissible even when the parameter estimate is affected by measurement noise.

*Table 7.* Source recovery for the 2D Helmholtz equation with $k = 1$, reporting relative source errors and boundary constraint residuals for different sensor counts and noise levels.

| | | LM-PIELM | | LM-PIELM (AUG.) | |
|---|---|---|---|---|---|
| $M$ | $\eta$ | $\frac{\|f_{\text{ex}} - \hat{f}\|_2}{\|f_{\text{ex}}\|_2}$ | $\|\Phi_b \mathbf{C} - \mathbf{g}\|_\infty$ | $\frac{\|f_{\text{ex}} - \hat{f}\|_2}{\|f_{\text{ex}}\|_2}$ | $\|\Phi_b \mathbf{C} - \mathbf{g}\|_\infty$ |
| 50 | 0% | $1.89\times10^{-1}$ | $5.20\times10^{-17}$ | $1.89\times10^{-1}$ | $7.09\times10^{-17}$ |
| 100 | 0% | $1.40\times10^{-3}$ | $1.15\times10^{-16}$ | $1.40\times10^{-3}$ | $1.32\times10^{-16}$ |
| 200 | 0% | $2.76\times10^{-5}$ | $1.06\times10^{-16}$ | $2.76\times10^{-5}$ | $3.08\times10^{-16}$ |
| 500 | 0% | $4.17\times10^{-8}$ | $2.25\times10^{-16}$ | $2.94\times10^{-8}$ | $3.10\times10^{-16}$ |
| 1000 | 0% | $1.37\times10^{-8}$ | $9.35\times10^{-16}$ | $8.92\times10^{-9}$ | $8.57\times10^{-16}$ |
| 50 | 1% | $7.07\times10^{-1}$ | $5.58\times10^{-17}$ | $7.07\times10^{-1}$ | $9.91\times10^{-17}$ |
| 100 | 1% | $3.36\times10^{-1}$ | $7.44\times10^{-17}$ | $3.36\times10^{-1}$ | $2.63\times10^{-16}$ |
| 200 | 1% | $2.79\times10^{-1}$ | $1.18\times10^{-16}$ | $2.79\times10^{-1}$ | $1.98\times10^{-16}$ |
| 500 | 1% | $2.16\times10^{-1}$ | $3.86\times10^{-16}$ | $2.16\times10^{-1}$ | $7.31\times10^{-16}$ |
| 1000 | 1% | $3.11\times10^{-1}$ | $8.30\times10^{-16}$ | $3.11\times10^{-1}$ | $6.14\times10^{-16}$ |
| 50 | 5% | $1.85\times10^{0}$ | $3.14\times10^{-17}$ | $1.85\times10^{0}$ | $1.13\times10^{-16}$ |
| 100 | 5% | $9.78\times10^{-1}$ | $9.05\times10^{-17}$ | $9.78\times10^{-1}$ | $6.79\times10^{-17}$ |
| 200 | 5% | $1.24\times10^{0}$ | $1.55\times10^{-16}$ | $1.24\times10^{0}$ | $2.19\times10^{-16}$ |
| 500 | 5% | $1.80\times10^{0}$ | $2.95\times10^{-16}$ | $1.80\times10^{0}$ | $3.89\times10^{-16}$ |
| 1000 | 5% | $1.04\times10^{0}$ | $9.12\times10^{-16}$ | $1.04\times10^{0}$ | $4.34\times10^{-16}$ |

*Table 8.* Source recovery for the 2D Helmholtz equation with $k = \pi$, reporting relative source errors and boundary constraint residuals for different sensor counts and noise levels.

| | | LM-PIELM | | LM-PIELM (AUG.) | |
|---|---|---|---|---|---|
| $M$ | $\eta$ | $\frac{\|f_{\text{ex}} - \hat{f}\|_2}{\|f_{\text{ex}}\|_2}$ | $\|\Phi_b \mathbf{C} - \mathbf{g}\|_\infty$ | $\frac{\|f_{\text{ex}} - \hat{f}\|_2}{\|f_{\text{ex}}\|_2}$ | $\|\Phi_b \mathbf{C} - \mathbf{g}\|_\infty$ |
| 50 | 0% | $1.86\times10^{-1}$ | $5.43\times10^{-17}$ | $1.86\times10^{-1}$ | $1.09\times10^{-16}$ |
| 100 | 0% | $1.96\times10^{-3}$ | $1.27\times10^{-16}$ | $1.96\times10^{-3}$ | $1.70\times10^{-16}$ |
| 200 | 0% | $1.70\times10^{-6}$ | $1.61\times10^{-16}$ | $1.86\times10^{-6}$ | $2.67\times10^{-16}$ |
| 500 | 0% | $3.37\times10^{-8}$ | $4.26\times10^{-16}$ | $3.77\times10^{-8}$ | $1.89\times10^{-16}$ |
| 1000 | 0% | $2.52\times10^{-8}$ | $7.21\times10^{-16}$ | $1.47\times10^{-8}$ | $3.68\times10^{-16}$ |
| 50 | 1% | $9.45\times10^{-1}$ | $7.20\times10^{-17}$ | $9.45\times10^{-1}$ | $1.09\times10^{-16}$ |
| 100 | 1% | $6.96\times10^{-1}$ | $4.99\times10^{-17}$ | $6.96\times10^{-1}$ | $1.34\times10^{-16}$ |
| 200 | 1% | $6.51\times10^{-1}$ | $1.24\times10^{-16}$ | $6.51\times10^{-1}$ | $1.80\times10^{-16}$ |
| 500 | 1% | $5.17\times10^{-1}$ | $6.06\times10^{-16}$ | $5.17\times10^{-1}$ | $3.99\times10^{-16}$ |
| 1000 | 1% | $4.51\times10^{-1}$ | $1.07\times10^{-15}$ | $4.51\times10^{-1}$ | $6.86\times10^{-16}$ |
| 50 | 5% | $2.75\times10^{0}$ | $7.76\times10^{-17}$ | $2.75\times10^{0}$ | $6.92\times10^{-17}$ |
| 100 | 5% | $2.02\times10^{0}$ | $4.65\times10^{-17}$ | $2.02\times10^{0}$ | $2.84\times10^{-16}$ |
| 200 | 5% | $2.49\times10^{0}$ | $1.56\times10^{-16}$ | $2.49\times10^{0}$ | $2.55\times10^{-16}$ |
| 500 | 5% | $2.34\times10^{0}$ | $5.73\times10^{-16}$ | $2.34\times10^{0}$ | $5.06\times10^{-16}$ |
| 1000 | 5% | $1.80\times10^{0}$ | $1.15\times10^{-15}$ | $1.80\times10^{0}$ | $3.17\times10^{-16}$ |

### B.2. Source Recovery in the 2D Helmholtz Equation

We consider source recovery for $u_{xx} + u_{yy} + k^2 u = f$ on $(0,1)^2$ with homogeneous Dirichlet conditions. The exact solution is $u = \sin(\pi x)\sin(\pi y)$, so $f = (k^2 - 2\pi^2)\sin(\pi x)\sin(\pi y)$. We use $N = 225$ ($15 \times 15$, degree 14), $N_c = 7396$, $N_b = 348$, and a test set of 24 500 points, including 2000 boundary points. Results are reported for $k = 1$ and $k = \pi$. For LM-PIELM, the Tikhonov parameter is $\beta = 10^{-10}$ in the noiseless setting and $\beta = 10^{-2}$ under noise. For LM-PIELM (Aug.), we use $\beta = 10^{-10}$ in the noiseless setting and $\beta = 10^{-2}$ under noise. The larger noisy-case regularisation is used only for the source-recovery least-squares block. The boundary constraints remain hard constraints in both forms. The solve time ranges from 0.04-0.08 s for LM-PIELM and 0.23-1.7 s for LM-PIELM (Aug.), increasing with sensor count.

Tables 7 and 8 show the full sensor and noise sweep for $k = 1$ and $k = \pi$. In the noiseless setting, source recovery improves rapidly with $M$, reaching errors of order $10^{-8}$ at $M = 1000$ for both wavenumbers. Under noise, the recovery problem becomes much more sensitive because the source is obtained by applying a differential operator to the reconstructed solution. The augmented form is more robust in these noisy cases. For example, at $k = 1$ and 5% noise, source error for both LM-PIELM and LM-PIELM (Aug.) stays around 1.0-1.9. At $k = \pi$, the same trend is observed, with larger errors because of the reaction term.

The important point is that the boundary behaviour does not follow the source-recovery error. In both tables, the boundary

residual remains at $10^{-15}$-$10^{-17}$ for LM-PIELM and at $10^{-16}$-$10^{-17}$ for LM-PIELM (Aug.), across all sensor counts, noise levels, and wavenumbers. This again shows that the algebraic boundary constraints are preserved independently of the quality of the recovered interior source.