2026 European Conference on Computing in Construction
Corfu, Greece
July 12–15, 2026

Parametric VPINN Framework for Rapid Design Exploration of Thin Slabs

Ahmad Arafat[1,3], Dimosthenis Kifokeris[1,2], and Mohamad Omar Alzokani[1]
[1]Dynzatec AB, Gothenburg, Sweden
[2]Chalmers University of Technology, Gothenburg, Sweden
[3]Etteplan Sweden AB, Gothenburg, Sweden

Abstract

This paper proposes a robust Variational Physics-Informed Neural Network (VPINN) framework specifically tailored for the structural analysis of thin Kirchhoff-Love plates. We utilize Approximate Distance Functions (ADFs) for hard boundary constraints and a hybrid spectral variational formulation to overcome challenges associated with fourth-order partial differential equations, such as vanishing gradients and the need for C^1-continuous meshing. Our VPINN framework can learn a parametric design space and enable real-time, mesh-free structural analysis, eliminating the ''modeling-meshing-solving'' cycle. The results confirm that our VPINN accurately captures critical stiffness trends and global deformation modes, satisfying the precision requirements for conceptual design and topology optimization.

Introduction: Background, related literature, and research aim

Structural engineering plays a pivotal role in defining parameters of safety, material efficiency, and decision-making processes in building projects (Ketabi and Heravi, 2022). Among structural components, slabs often serve as the primary horizontal load-bearing elements. The vast majority of slabs in residential and commercial buildings are geometrically characterized as ''thin plates'' (span-to-depth ratio $L/h > 20$), where the structural response is dominated by bending, and transverse shear deformations are negligible (Timoshenko and Woinowsky-Krieger, 1959). Consequently, the Kirchhoff-Love plate theory, which neglects shear deformation, serves as the most appropriate mechanical description for these elements (Marti et al., 2022). While the Mindlin-Reissner theory is essential for thick plates ($L/h < 10$) to account for shear flexibility, its application to thin slabs introduces unnecessary computational variables and complexity (Teixeira de Freitas et al., 2025). The Kirchhoff-Love theory allows for more computationally efficient models accurately reflecting the stiffer behavior of thin slabs (Marti et al., 2022), potentially leading to more resource-efficient material usage during building production.

The Finite Element Method (FEM) is widely adopted to analyze the structural behavior of slabs. However, despite its high fidelity, FEM faces significant limitations in the context of rapid, iterative design; its reliance on mesh generation becomes a bottleneck during design optimization loops with frequent geometry changes (Park and Kang, 2024). Moreover, most commercial FEM software utilizes Mindlin-Reissner formulations during FEM (using C^0-continuous elements) to avoid the numerical difficulties associated with Kirchhoff-Love theory; implementing Kirchhoff-Love in FEM would require the displacement field to be C^1-continuous (smooth slope everywhere), necessitating complex and computationally expensive elements, such as Hermite or Argyris (Zienkiewicz and Taylor, 2000). However, the generalized utilization of Mindlin-Reissner in FEM unavoidably introduces, in the case of thin slabs, the computational inefficiencies mentioned earlier.

Recently, Physics-Informed Neural Networks (PINNs) have emerged as a promising alternative for solving Partial Differential Equations (PDEs) (Raissi et al., 2019). Unlike FEM, PINNs are mesh-free and differentiable, allowing them to learn the solution operator over a continuous parametric space. This can enable a single trained network to predict the structural response across thousands of design variations instantly, effectively replacing the time-consuming ''modeling-meshing-solving'' cycle with real-time inference (Raissi et al., 2019). However, standard PINNs relying on the strong form of the governing equations struggle with the fourth-order derivatives present in the Kirchhoff-Love bi-harmonic equation ($\nabla^4 w = q/D$); the computation of high-order derivatives via automatic differentiation often leads to numerical instability and vanishing gradients, Wang et al. (2021a).

To address this, two variational approaches that reduce the order of differentiation have gained traction: the Deep Energy Method (DEM), based on the Ritz method (minimizing potential energy) (Samaniego et al., 2020), and the Variational Physics-Informed Neural Network (VPINN), based on the Galerkin method (minimizing the residual of the variational form) (Kharazmi et al., 2021). DEM is intuitive, but recent studies suggest that its unconstrained loss formulation can exhibit instability when learning across complex geometric parameter spaces due to spectral biases in the energy landscape (Wang et al., 2021b). In contrast, the Galerkin-based VPINN framework offers a robust mechanism to project residuals onto test functions, providing improved stability for parametric problems (Kumar and Singh, 2025). In particular, Huang and Peng (2024) proposed an improved plate DEM utilizing finite differ-

ence approximations via convolutional kernels and domain decomposition with coordinate transformations for irregular Kirchhoff plates, achieving high pointwise accuracy on L-shaped geometries. However, their method was demonstrated only for single-instance configurations with structured grids, whereas the parametric setting, where a single network must generalize across a continuous family of geometries, remains largely unexplored.

Therefore, for thin slabs, leveraging VPINN can take advantage of the suitability of the Kirchhoff-Love theory while addressing the combination of computational inefficiencies discussed above; however, such a solution has received limited attention in the literature. As such, this paper proposes a robust framework for VPINN specifically tailored for the analysis of thin slabs. By utilizing the variational form of Kirchhoff-Love theory, we reduce the derivative requirements to the second order, ensuring numerical stability while maintaining mechanical consistency. We demonstrate the framework's capability to learn a parametric design space and discuss its potential to enable real-time, mesh-free structural analysis.

Kirchhoff-Love plate theory

The mechanical behavior of the slab is modeled using the Kirchhoff-Love plate theory, which provides the governing mathematical description for thin structural elements where the span-to-thickness ratio is sufficiently large ($L/h \geq 20$) (Timoshenko and Woinowsky-Krieger, 1959). This theory relies on the kinematic assumption that straight lines remain inextensible and normal to the mid-surface before and after deformation. Consequently, transverse shear strains are neglected, and the deformation is fully characterized by the transverse displacement $w(x, y)$ of the mid-surface $\Omega \subset \mathbb{R}^2$. Under the assumption of linear isotropic elasticity and small deformations, the relationship between the bending moments M and the generalized curvatures $\boldsymbol{\kappa}$ is governed by the constitutive equation $\mathrm{M} = \mathrm{D}\boldsymbol{\kappa}$ (Reddy, 2006).

The equilibrium of the plate under a distributed transverse load q is described by the strong form of the governing partial differential equation. By combining the kinematic relations, constitutive laws, and equilibrium conditions, the problem reduces to the classical bi-harmonic equation, i.e., a fourth-order partial differential equation:

$$D\nabla^4 w = D\left(\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4}\right) = q \quad \text{in } \Omega, \quad (1)$$

where ∇^4 denotes the bi-harmonic operator and D is the flexural rigidity of the plate, defined as:

$$D = \frac{Eh^3}{12(1-\nu^2)}, \quad (2)$$

dependent on the Young's modulus E, Poisson's ratio ν, and the slab thickness h.

To form a well-posed problem, Eq. (1) must be complemented by boundary conditions. In this study, we focus on clamped edges, which are representative of slabs rigidly connected to stiff vertical supports (e.g., shear walls or monolithic columns). For clamped edges (Γ_C), the boundary conditions are defined as follows:

$$\Gamma_C : \left.\begin{aligned} w &= 0 \\ \frac{\partial w}{\partial n} &= 0 \end{aligned}\right\}, \quad (3)$$

where n denotes the outward unit normal direction.

To avoid the numerical instability of computing fourth-order derivatives in PINNs (Wang et al., 2021a), we adopt the variational formulation. Multiplying Eq. (1) by a test function $v \in \mathcal{V}$ satisfying homogeneous essential boundary conditions and integrating by parts twice yields:

$$\int_\Omega D\left[\nabla^2 w \nabla^2 v - (1-\nu)\mathcal{K}(w, v)\right] d\Omega = \int_\Omega qv \, d\Omega, \quad (4)$$

where $\mathcal{K}(w, v)$ represents the Gaussian curvature interaction term:

$$\mathcal{K}(w, v) = \frac{\partial^2 w}{\partial x^2}\frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 w}{\partial y^2}\frac{\partial^2 v}{\partial x^2} - 2\frac{\partial^2 w}{\partial x \partial y}\frac{\partial^2 v}{\partial x \partial y}. \quad (5)$$

Although $\int \mathcal{K} d\Omega$ theoretically vanishes for clamped slabs, we retain the full formulation for generality. Minimizing this variational residual requires only second-order derivatives, significantly enhancing numerical stability compared to strong-form collocation methods.

Approximate Distance Functions (ADFs)

Standard PINNs enforce boundary conditions via penalty terms, which introduce competing objectives and often lead to inexact satisfaction for fourth-order problems. To overcome this, we employ Approximate Distance Functions (ADFs) (Sukumar and Srivastava, 2021) for ''hard constraint'' enforcement. In this framework, the neural network output $\hat{w}(\mathrm{x})$ is constructed not as a direct approximation of the displacement, but as a composite ansatz:

$$\hat{w}(\mathrm{x}) = \Phi(\mathrm{x}) \cdot \mathcal{N}(\mathrm{x}; \theta), \quad (6)$$

where $\mathcal{N}(\mathrm{x}; \theta)$ is the raw output of the deep neural network parameterized by weights θ, and $\Phi(\mathrm{x})$ is a geometrically constructed distance function that strictly encodes the boundary requirements.

Theory of R-Functions

The function $\Phi(\mathrm{x})$ is constructed using the theory of R-functions (real-valued logic functions), which allows for the composition of complex domain geometries from simple primitives (e.g., lines, circles) using Boolean operations. For a domain Ω formed by the intersection or union of N primitive shapes, the global distance function $\phi(\mathrm{x})$ is derived using the R-equivalence operation:

$$\phi(\mathrm{x}) = \mathcal{R}_m(\phi_1, \phi_2, \ldots, \phi_N), \quad (7)$$

where ϕ_i are the distance functions of the individual primitives. The R-equivalence operation ensures smoothness

(C^m-continuity) and gradient normalization ($|\nabla\phi| = 1$ near the boundary), preventing gradient pathologies during optimization (Biswas et al., 2004).

Application to clamped slabs

For the specific case of clamped Kirchhoff-Love plates, the boundary conditions on the clamped edge Γ_C require both the deflection and the normal derivative to vanish ($w = 0$ and $\partial w/\partial n = 0$). To satisfy both conditions simultaneously using the ADFs, the geometric ansatz in Eq. (6) is modified to use the square of the distance function:

$$\Phi(\mathrm{x}) = [\phi(\mathrm{x})]^2. \tag{8}$$

Since $\phi(\mathrm{x}) = 0$ on Γ_C, it follows that $\Phi(\mathrm{x}) = 0$. Furthermore, the derivative of the ansatz is $\nabla\Phi = 2\phi\nabla\phi$, which also vanishes on the boundary where $\phi = 0$. Consequently, the trial solution $\hat{w} = \phi^2\mathcal{N}$ automatically satisfies:

$$\hat{w}|_{\Gamma_C} = 0 \quad \text{and} \quad \left.\frac{\partial\hat{w}}{\partial n}\right|_{\Gamma_C} = 0, \tag{9}$$

regardless of the values predicted by $\mathcal{N}$. This eliminates the need for boundary loss terms, reducing optimization to a purely physical energy minimization problem.

VPINN framework for parametric slab analysis

In this section, we present the details of the VPINN framework as sketched in Fig. 2.

Geometric parametrization and input space

The structural component under investigation is a non-convex, L-shaped slab, a geometry frequently encountered in architectural layouts around elevator cores or re-entrant corners (Timoshenko and Woinowsky-Krieger, 1959). To establish a comprehensive parametric design space, the geometry is driven by two independent dimensionless parameters, $\mathcal{P} = \{R_1, R_2\}$, which control the aspect ratios of the horizontal and vertical slab segments, respectively.
The physical dimensions of the slab are derived from these parameters to ensure that the domain remains within a normalized bounding box. As depicted in Fig. 1, the total height (b_1) is normalized to unity (1.0) and the top leg width (a_2) is fixed at 0.5. The remaining dimensions are directly controlled by the parameters, such that the total width is $a_1 = b_1/R_1$ and the right leg height is $b_2 = a_2/R_2$.
The parameters R_1 and R_2 vary continuously within the range $[1.0, 1.6]$, creating a family of geometries ranging from thick, compact L-shapes to slender, extended configurations. A critical geometric feature implemented in this framework is the introduction of a circular fillet normalized radius of $r = 0.05$ at the re-entrant corner (the inner corner of the L-shape). This modification removes the geometric singularity associated with sharp re-entrant corners. As noted by Sukumar and Srivastava (2021), smoothing re-entrant corners is essential when using distance-based ansatz functions, as it ensures the gradient of the distance function remains well-defined and continuous ($\nabla\phi \in C^0$) along the entire boundary.

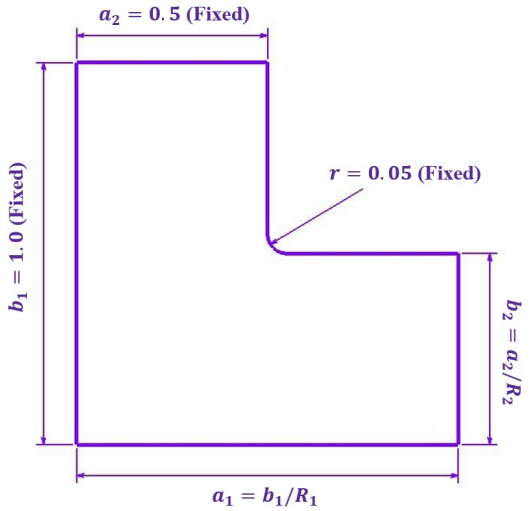


Figure 1: Parametric L-shaped slab geometry

Consequently, the neural network input space is four-dimensional, consisting of the spatial coordinates and the design parameters: $\mathrm{x}_{in} = \{x, y, R_1, R_2\}$.

Neural network architecture in the VPINN

To approximate the solution map $\mathcal{N}(\mathrm{x}_{in}; \theta)$, we employ a fully connected feed-forward neural network, commonly referred to as a Multi-Layer Perceptron (MLP), which serves as the core approximator within the VPINN framework. The architecture is designed to map the four-dimensional input vector $\mathrm{x}_{in} = \{x, y, R_1, R_2\}$ to the scalar latent deflection field w_{model}.
The network depth and width are configured to ensure sufficient expressive capacity for learning the complex solution manifold across the parametric design space. The architecture consists of 6 hidden layers, with each layer containing 512 neurons. This results in a deep over-parameterized model capable of learning the complex parametric solution manifold. To facilitate the training of this deep architecture and prevent the degradation problem (vanishing gradients), residual skip connections are implemented across the hidden layers (He et al., 2016). Mathematically, the output of the l-th layer $\mathrm{z}^{(l)}$ is given by:

$$\mathrm{z}^{(l)} = \sigma(\mathrm{W}^{(l)}\mathrm{z}^{(l-1)} + \mathrm{b}^{(l)}) + \mathrm{z}^{(l-1)}, \tag{10}$$

where W and b are the weights and biases, and σ is the activation function.
A critical feature of this implementation is the use of the Self-Scalable Tanh (Stan) activation function. Unlike standard fixed activation functions (e.g., ReLU or Tanh), the Stan function includes a single global learnable scalar parameter β, shared across all layers of the network, that allows the model to adaptively scale the slope of the activation during training. The function is defined as:

$$\sigma(z) = \tanh(z) \cdot (1 + \beta z). \tag{11}$$

As demonstrated by Gnanasambandam et al. (2023), this specific formulation provides a smooth, non-saturating response. By optimizing a single global scaling factor β, the network can automatically tune the aggregate spectral properties of the approximation without the computational overhead or optimization complexity of layer-wise adaptation. Furthermore, Weight Normalization is applied to all linear layers to decouple the magnitude of the weight vectors from their direction, further stabilizing the optimization process (Salimans and Kingma, 2016).

Ansatz construction for hard boundary constraints

A defining characteristic of the proposed framework is the exact satisfaction of boundary conditions through a geometric ansatz, rather than relying on penalty terms in the loss function. To achieve this, we implement the hard constraint formulation derived in the previous section. The final physical deflection field, $\hat{w}(\mathrm{x})$, is constructed by modulating the raw scalar output of the neural network, $w_{model}(\mathrm{x})$, with the square of the approximate distance function $\phi(\mathrm{x})$:

$$\hat{w}(\mathrm{x}) = w_{model}(\mathrm{x}) \cdot [\phi(\mathrm{x})]^2. \tag{12}$$

Since $\phi = 0$ on Γ_C, both $\hat{w}$ and $\partial\hat{w}/\partial n$ vanish automatically, confining the optimization to kinematically admissible functions and eliminating boundary penalty terms (Berg and Nyström, 2018; Sukumar and Srivastava, 2021).

Construction of test functions

In the VPINN framework, the choice of the test space $\mathcal{V}$ is critical for the accuracy and stability of the numerical integration. While traditional formulations often rely solely on polynomial bases (Kharazmi et al., 2021), we adopt a hybrid spectral approach to capture a broader range of solution frequencies. Specifically, we construct the test space by combining Legendre polynomials, Chebyshev polynomials, and trigonometric functions. This leverages the orthogonality and polynomial approximation power in Legendre and Chebyshev (Canuto et al., 2006) while incorporating the periodic spectral properties of trigonometric series to address the ''spectral bias'' inherent in fully connected neural networks (Rahaman et al., 2019), thereby enhancing the network's ability to resolve complex local features.

The Galerkin formulation requires test functions to satisfy homogeneous essential boundary conditions ($v = 0$, $\nabla v \cdot \mathrm{n} = 0$ on Γ_C) (Hughes, 2012). We enforce this via the same geometric masking used for the trial ansatz (Sukumar and Srivastava, 2021). Let $B_k(\mathrm{x})$ denote the k-th order raw basis function (selected from the combined pool of Legendre, Chebyshev, or trigonometric functions) evaluated at coordinate x. The admissible test function $v_k(\mathrm{x})$ is constructed as:

$$v_k(\mathrm{x}) = B_k(\mathrm{x}) \cdot [\phi(\mathrm{x})]^2, \tag{13}$$

where $\phi(\mathrm{x})$ is the same approximate distance functions (ADFs) used for the trial ansatz. By modulating the hybrid basis with the squared distance field, we guarantee that every test function in the discrete space $\mathcal{V}_h$ vanishes along with its normal derivative at the boundary. This allows for a consistent mesh-free implementation where the variational form residuals are computed solely via domain integration, avoiding the numerical complexity of boundary integrals.

Variational loss formulation

The VPINN minimizes the residual of the integral variational form rather than pointwise strong-form residuals. Since all boundary integrals vanish due to the hard constraints on both $\hat{w}$ and v_k, the variational residual for the k-th test function reduces to:

$$\begin{aligned}\mathcal{R}_k(\theta) = \int_\Omega D\Big[& (\hat{w}_{xx} v_{k,xx} + \hat{w}_{yy} v_{k,yy}) \\ & + \nu(\hat{w}_{xx} v_{k,yy} + \hat{w}_{yy} v_{k,xx}) \\ & + 2(1-\nu)\hat{w}_{xy} v_{k,xy} \Big] d\Omega \\ & - \int_\Omega q v_k d\Omega.\end{aligned} \tag{14}$$

To numerically evaluate these integrals, we employ Monte Carlo integration over the sampled interior points. The total loss function $\mathcal{L}(\theta)$ is constructed by aggregating the squared residuals over the finite set of N_v test functions:

$$\mathcal{L}(\theta) = \sum_{k=1}^{N_v} |\mathcal{R}_k(\theta)|^2 . \tag{15}$$

By minimizing this scalar loss, the optimizer forces our VPINN to satisfy the principle of virtual work over the entire parametric domain. This formulation is purely physics-based and unsupervised, requiring no labeled data or reference solutions (Kharazmi et al., 2021).

Results and discussion

To evaluate the predictive capability of the proposed parametric VPINN framework, we conduct a comparative analysis against high-fidelity FEM reference solutions generated using the legacy FEniCS finite element library with the Ciarlet-Raviart mixed formulation, which decomposes the biharmonic equation into two coupled second-order equations. The FEM discretization employs quadratic Lagrange elements (P_2) on unstructured triangular meshes with approximately 236,440 vertices per configuration, with mesh convergence verified to within 0.5% variation in maximum deflection. For the training phase, the spatial and parametric domains were discretized using 80,000 collocation points generated via uniform quasirandom sampling, ensuring efficient coverage of the irregular L-shaped geometry. The network parameters were optimized using the L-BFGS algorithm in a single optimization step containing approximately 300,000 internal line-search iterations, following the PhysicsNeMo single-step pattern. The optimization process exhibited robust convergence, reducing the variational residual loss from an initial magnitude

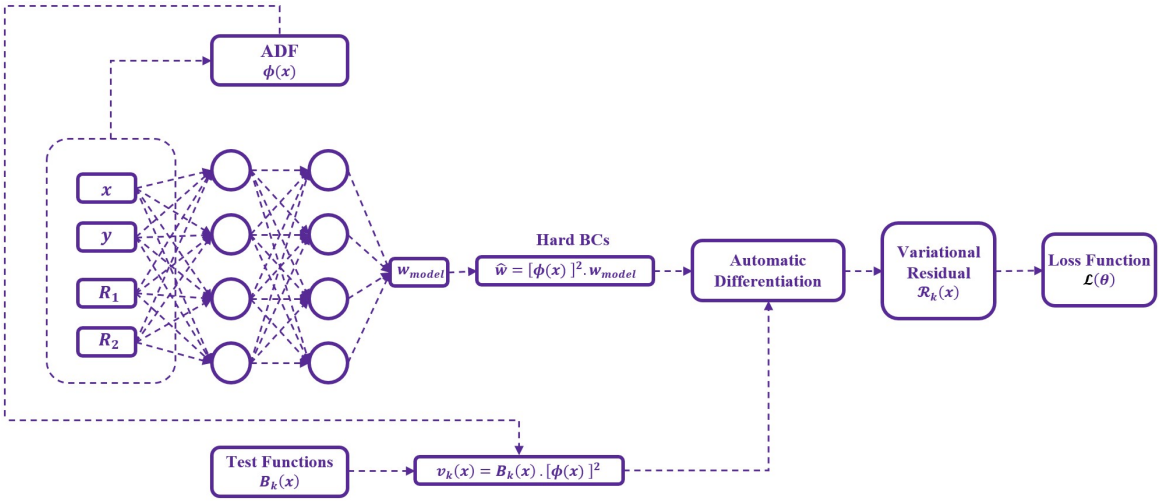


Figure 2: Schematic diagram of the proposed VPINN

of $O(10^1)$ to a final stable residual of $O(10^{-6})$, confirming that the network successfully located a distinct energy minimum.

For the numerical experiments, the physical problem is fully non-dimensionalized. The flexural rigidity is normalized to $D = 1$, the uniform transverse load is set to $q = -1$, and the Poisson's ratio is fixed at $\nu = 0.3$. Furthermore, the spatial domain size is non-dimensionalized such that the entire geometry is strictly bounded within the unit square defined by coordinates $(0, 0)$ and $(1, 1)$. The total offline training cost was approximately 4 hours on a single NVIDIA A100 (80GB) GPU, with a peak memory consumption of approximately 79 GB. In contrast, each FEM reference solution required approximately 63 seconds at mesh resolution 400 (≈236,440 vertices). The amortized computational advantage of the trained VPINN becomes favorable beyond approximately $N_{\text{break}} \approx \lceil T_{\text{train}}/T_{\text{FEM}} \rceil \approx 230$ design evaluations, a threshold readily exceeded in optimization workflows requiring hundreds to thousands of iterations. The neural network was trained once over the parametric domain, and subsequently evaluated to predict the deflection fields for varying geometric configurations without any retraining. This inference process demonstrates true real-time capability, requiring approximately 0.015 seconds per design evaluation on an NVIDIA A100 (80GB) GPU.

Generalization across geometric parameters

We present inference results for two representative geometric cases that test the network's ability to generalize across both symmetric and asymmetric configurations:

1. **Case I:** $R_1 = 1.0, R_2 = 1.0$. This case represents the baseline geometry, characterized by a fully symmetric configuration with equal leg dimensions.

2. **Case II:** $R_1 = 1.0, R_2 = 1.6$. This case represents a boundary of the parametric design space, corresponding to a highly asymmetric configuration with a compressed right leg ($b_2 = 0.3125$). It validates that the VPINN has captured the independent sensitivity of the deflection field to each geometric parameter, rather than simply learning symmetric scaling.

Figure 3 illustrates the comparison between the predicted deflection field $\hat{w}_{\text{VPINN}}$ and the reference solution $\hat{w}_{\text{FEM}}$ for these configurations. A comprehensive quantitative evaluation across all 16 grid points in the parametric space (Fig. 4) is presented in the following subsection.

Error analysis and validation

The absolute error distributions (right column of Fig. 3) reveal the exact satisfaction of boundary conditions. The error vanishes completely at the clamped edges (Γ_C), empirically confirming the theoretical validity of the ansatz construction $\hat{w} = w_\theta \cdot \phi^2$. The VPINN correctly identifies the shift in the maximum deflection point in the asymmetric Case II, demonstrating that the network has effectively learned the parametric dependence of the solution map.

To comprehensively evaluate the framework's generalization across the design space, we conduct a systematic grid evaluation over 16 configurations spanning $R_1, R_2 \in \{1.0, 1.2, 1.4, 1.6\}$. Figure 4 presents the global relative (L_2) error and maximum deflection relative ($\Delta w_{\max}$) error as heatmaps over the parametric grid. The model achieves a mean L_2 error of 7.87% (min: 5.75%, max: 11.55%) and a mean $\Delta w_{\max}$ of 5.90% (min: 0.68%, max: 11.49%) across all 16 cases. Error analysis reveals a systematic increase for larger R_1 values, corresponding to narrower bottom-leg geometries where the aspect ratio is most extreme. The outlier at $(R_1 = 1.6, R_2 = 1.2)$ represents the most elongated configuration, where the fixed-order test function basis provides reduced effective resolution. Notably, the maximum deflection error, the quantity most relevant for serviceability assessment, remains below 10% for 14 of 16 cases, confirming the model's suitability for rapid design screening. This error magnitude is consistent with serviceability tolerances in Eurocode 2 (EN 1992-1-1, $w_{\max} \leq L/250$) and the 10% threshold for successful PINN solutions established by the PINNacle bench-

mark (Hao et al., 2024).

Critically, the error distribution analysis challenges common assumptions regarding L-shaped domains. While the re-entrant corner typically dominates numerical error due to reduced regularity, our parametric filleting successfully regularizes this region. Consequently, maximum deviations shift to the outer clamped corners, where the strict enforcement of zero-slope constraints imposes high-frequency gradients.

This behavior highlights the specific interaction between the test space and the boundary constraints:

1. **Hybrid Spectral Resolution:** The use of a mixed test space combining Legendre polynomials and Chebyshev polynomials (Canuto et al., 2006) with trigonometric functions (Rahaman et al., 2019) was crucial for stabilizing the variational loss. While this hybrid basis effectively captures the global oscillatory modes of the plate, the high-frequency gradients required to strictly enforce the zero-slope condition ($\nabla w \cdot \mathrm{n} = 0$) at the sharp outer corners remain computationally demanding to resolve with high precision. The systematic error increase observed for larger R_1 values in Fig. 4 confirms that the fixed-order basis provides non-uniform resolution across geometries with extreme aspect ratios.

2. **Trade-off against Coordinate Mapping:** In classical spectral methods, achieving high-precision convergence on irregular domains with corner singularities typically requires coordinate mapping or domain decomposition to regularize the geometry (Boyd, 2001). However, applying such mappings in a parametric learning framework is computationally prohibitive. A continuous parametric mapping would require the computation of parameter-dependent Jacobians at every collocation point, significantly increasing the training cost and architectural complexity. Consequently, our Cartesian approach represents a deliberate trade-off: accepting a bounded approximation error (mean L_2 of 7.87% across 16 configurations) to enable a unified, mapping-free parametric model that enables real-time inference.

3. **Design-Centric Accuracy:** In the context of inverse design and topology optimization, the primary requirement is the accurate prediction of stiffness trends and relative performance between different geometries, rather than the recovery of the exact solution to floating-point precision. As evidenced by the systematic evaluation across 16 configurations (Fig. 4), our VPINN consistently captures the correct global deformation mode and parametric sensitivity, with a mean maximum deflection error of 5.90%. This confirms that the model has learned the parametric physics required to drive an optimization loop across the entire design space.

Conclusions

This study has developed and validated a robust parametric VPINN framework specifically tailored for the structural analysis of thin Kirchhoff-Love plates. By integrating ADFs for hard boundary constraints with a hybrid spectral variational formulation, the proposed method successfully overcomes the traditional challenges associated with fourth-order partial differential equations, such as vanishing gradients and the need for C^1-continuous meshing.

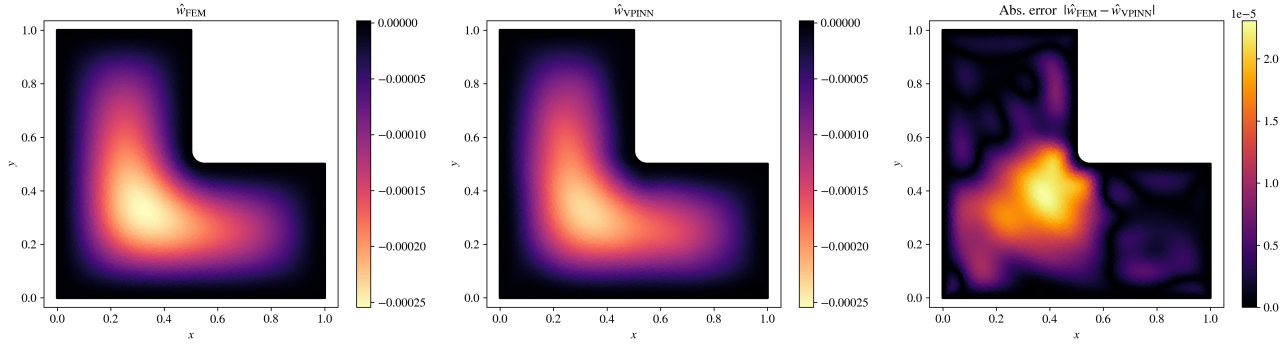
The scientific and practical implications of this work are summarized as follows:

1. **Robustness of the Variational Formulation:** The utilization of a hybrid test space (combining Legendre polynomials, Chebyshev polynomials, and trigonometric functions) proved essential for numerical stability. The framework achieved robust convergence, reducing the variational energy residual from $O(10^1)$ to $O(10^{-6})$. This confirms that limiting the derivative requirements to the second order via the variational form is a superior strategy for bending-dominated problems compared to strong-form collocation.

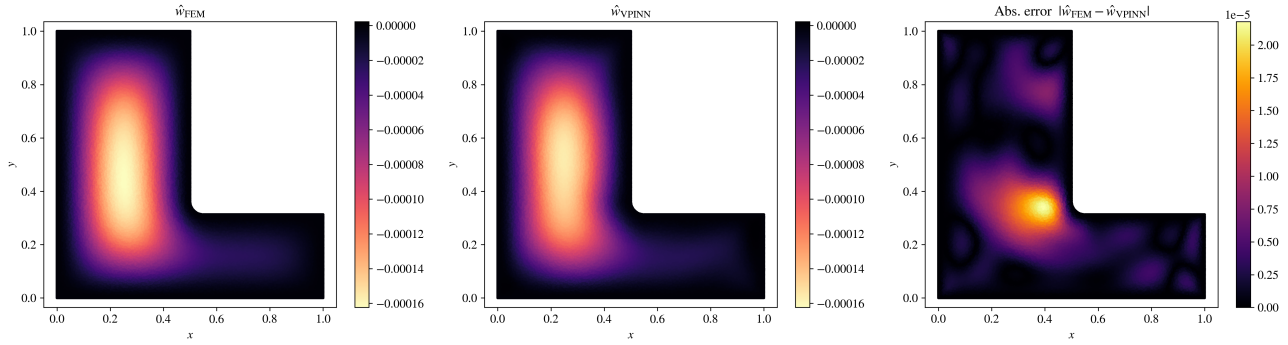
2. **Exact Boundary Adherence:** The implementation of the geometric ansatz $\hat{w} = w_\theta \cdot \phi^2$ guaranteed the exact satisfaction of clamped boundary conditions, as empirically confirmed by vanishing error at all clamped edges across the 16 evaluated configurations. This eliminates the need for hyperparameter tuning of boundary penalty terms, a common bottleneck in standard PINN architectures, and ensures kinematic admissibility regardless of the network's training state.

3. **Parametric Efficiency vs. Precision Trade-off:** Systematic evaluation across 16 configurations spanning the $R_1 \times R_2$ design space demonstrated a mean global L_2 error of 7.87% and a mean maximum deflection error of 5.90%, with 14 of 16 cases remaining below 10%. While higher than refined FEM, this represents a strategic trade-off. By avoiding computationally expensive coordinate mappings, our VPINN framework delivers a unified, mapping-free parametric model that accurately captures critical stiffness trends and global deformation modes, satisfying the precision requirements for conceptual design and topology optimization.

4. **Real-Time Design Capability:** Once trained (approximately 4 hours on a single NVIDIA A100 GPU), the VPINN enables inference of mechanical fields in approximately 0.015 seconds per design, amortizing the training investment after approximately 230 evaluations compared to FEM. This eliminates the classical ''modeling-meshing-solving'' cycle and paves the way for integrating physics-based surrogate models into real-time design workflows, allowing struc-



(a) Case I – $R_1 = 1.0, R_2 = 1.0$



(b) Case II – $R_1 = 1.0, R_2 = 1.6$

Figure 3: Comparative analysis of FEM reference (left), VPINN prediction (center), and absolute error (right)

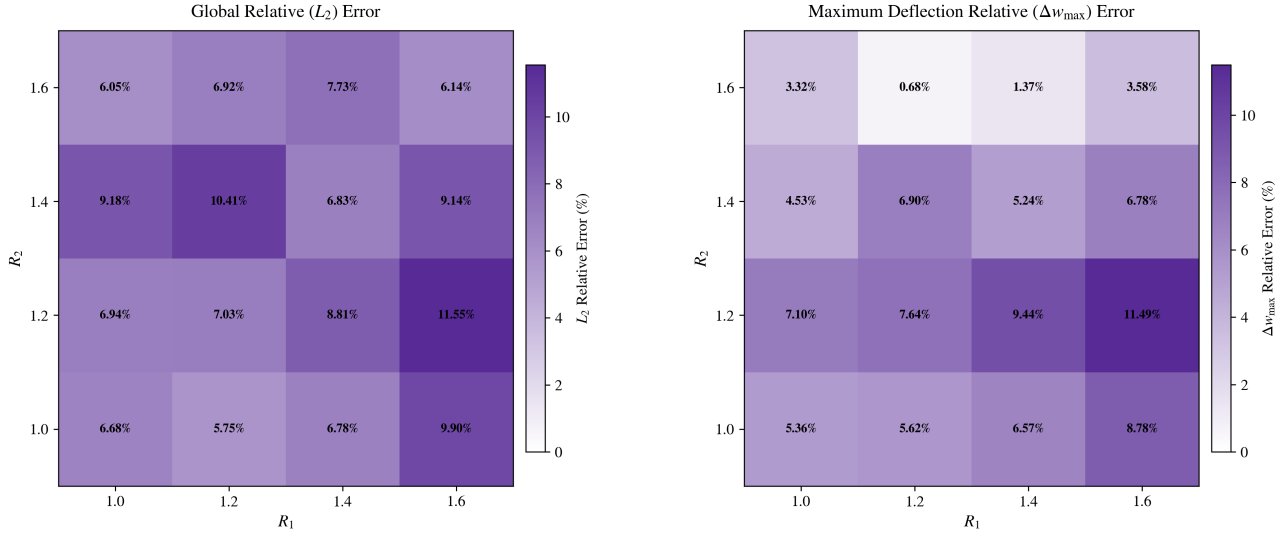


Figure 4: Parametric error distribution across the $R_1 \times R_2$ design space (16 validation cases). L_2 error: mean = 7.87%, min = 5.75%, max = 11.55%. $\Delta w_{\max}$ error: mean = 5.90%, min = 0.68%, max = 11.49%.

tural engineers to explore thousands of design iterations with immediate structural feedback.

Future work will extend this framework along three directions: (i) expanding the parametric space to include slab thickness, load distributions, and mixed boundary conditions; (ii) benchmarking against data-driven surrogate approaches to quantify the advantages of physics-based training; and (iii) incorporating reinforced concrete nonlinearities, including cracking and steel reinforcement, to bridge the gap between conceptual design and detailed verification.

References

Berg, J. and Nyström, K. (2018). A unified deep artificial neural network approach to partial differential equations in complex geometries. Neurocomputing, 317:28–41.

Biswas, A., Shapiro, V., and Tsukanov, I. (2004). Approximate distance functions. International Journal of Computational Geometry & Applications, 14(01n02):95–114.

Boyd, J. P. (2001). Chebyshev and Fourier Spectral Methods. Dover Publications, Mineola, New York, second edition.

Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. (2006). Spectral Methods: Fundamentals in Single Domains. Springer Science & Business Media, Berlin.

Gnanasambandam, R., Shen, B., Chung, J., Yue, X., and Kong, Z. (2023). Self-scalable tanh (stan): Faster convergence and better generalization in physics-informed neural networks. IEEE Transactions on Neural Networks and Learning Systems, 34(9):5687–5701.

Hao, Z., Yao, J., Su, C., Su, H., Wang, Z., Lu, F., Xia, Z., Zhang, Y., Liu, S., Lu, L., and Zhu, J. (2024). PINNacle: A comprehensive benchmark of physics-informed neural networks for solving PDEs. In Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778.

Huang, Z. and Peng, L. (2024). An improved plate deep energy method for the bending, buckling and free vibration problems of irregular Kirchhoff plates. Engineering Structures, 301:117235.

Hughes, T. J. R. (2012). The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover Publications, Mineola, New York. Section 5.3: Plate Bending.

Ketabi, A. B. and Heravi, G. (2022). Developing a framework for evaluating construction project safety levels and optimal cost allocation to safety influential factors. Engineering, Construction and Architectural Management, 29(9):643–668.

Kharazmi, E., Zhang, Z., and Karniadakis, G. E. (2021). Variational physics-informed neural networks for solving partial differential equations. Computer Methods in Applied Mechanics and Engineering, 374:113535.

Kumar, V. and Singh, G. (2025). A variational physics-informed neural network framework using petrov-galerkin method for solving singularly perturbed boundary value problems. arXiv preprint arXiv:2509.12271.

Marti, P., Kaufmann, W., Seelhofer, H., and Karagiannis, D. (2022). Kirchhoff–love plate deformations reinterpreted. Journal of Engineering Mechanics, 148(6):04022026.

Park, J. and Kang, N. (2024). Bmo-gnn: Bayesian mesh optimization for graph neural networks to enhance engineering performance prediction. Journal of Computational Design and Engineering, 11(1):260–271.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. (2019). On the spectral bias of neural networks. In Proceedings of the 36th International Conference on Machine Learning, pages 5301–5310. PMLR.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). 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.

Reddy, J. N. (2006). Theory and Analysis of Elastic Plates and Shells. CRC Press, Boca Raton, second edition.

Salimans, T. and Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. Advances in Neural Information Processing Systems, 29.

Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V., Guo, H., Hamdaoui, K., Zhuang, X., and Rabczuk, T. (2020). An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. Computer Methods in Applied Mechanics and Engineering, 362:112790.

Sukumar, N. and Srivastava, A. (2021). Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. Computer Methods in Applied Mechanics and Engineering, 389:114333.

Teixeira de Freitas, J. A., Tiago, C., and Pereira, E. M. B. R. (2025). Modelling of kirchhoff and mindlin–reissner plate bending with hybrid-trefftz stress elements. International Journal for Numerical Methods in Engineering, 126(3):e7668.

Timoshenko, S. P. and Woinowsky-Krieger, S. (1959). Theory of Plates and Shells. McGraw-Hill, New York, second edition.

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

Wang, S., Wang, H., and Perdikaris, P. (2021b). Learning the solution operator of parametric partial differential equations with physics-informed deeponets. Science Advances, 7(40):eabi8605.

Zienkiewicz, O. C. and Taylor, R. L. (2000). The Finite Element Method: Solid Mechanics. Butterworth-Heinemann, Oxford.