Solving the Biharmonic Equation with Coupled Physics-informed Neural Networks Based on Self-adaptive Loss Balancing

Jing Jiang
College of Mathematics and Statistics, Guangxi Normal University, Guilin 541006, China

Abstract: Standard Physics-informed neural networks (PINNs) encounter notable challenges in solving this equation numerically, such as inadequate approximation accuracy, considerable computational expense, and difficulties in loss function optimization. To mitigate these limitations, we propose introducing an auxiliary variable to decompose the fourth-order biharmonic equation into a coupled system of two second-order partial differential equations. By separately approximating the original solution and the intermediate variable, this approach effectively alleviates the numerical errors associated with computing high-order derivatives while also reducing computational overhead. Furthermore, we incorporate an adaptive loss weighting scheme to dynamically balance the contributions of the PDE various PDE loss terms during training, thereby addressing the issue of loss imbalance inherent in multi-constrained learning. The effectiveness of the proposed method is validated through several numerical experiments.

***Keywords*:** biharmonic equation, physics-informed neural networks, loss function, self-adaptive loss weighting

1. Introduction

The biharmonic equation, as a classic fourth-order partial differential equation [1], finds broad applications across a spectrum of engineering and physical disciplines, such as the Thin Plate Bending Problems of elasticity, Stokes flow of fluid mechanics, and potential field analysis in electromagnetism [2,3]. The essential challenge in solving this equation lies in accurately approximating its high-order derivatives while simultaneously satisfying complex combinations of boundary conditions, including coupled Dirichlet and Neumann constraints. Traditional numerical methods often exhibit significant limitations when addressing such demanding scenarios.

Traditional numerical approaches for solving the biharmonic equation mainly include the finite element method (FEM), the finite difference method (FDM), and the boundary element method (BEM) [4-6]. FDM relies on regular mesh discretization, necessitating mesh distortion correction for complex boundaries and thereby introducing the risk of additional numerical errors. FEM, though capable of adapting to complex geometries, involves laborious mesh generation and basis function construction, with computational costs increasing exponentially as the problem dimension and equation order rise. Although BEM reduces the dimensionality of the problem, it encounters significant difficulties in formulating boundary integral equations when dealing with nonlinear source terms. These methods are inherently mesh-dependent, which limits their flexibility and efficiency in highdimensional or complex boundary scenarios, making them less capable of meeting modern engineering demands for fast and accurate solutions.

The emergence of PINNs [7] has introduced a novel mesh-free paradigm for solving partial differential equations. By embedding physical laws into the loss function of neural networks, PINNs integrate data-driven learning with physical constraints to directly approximate the analytical solution, without relying on mesh generation — demonstrating notable advantages in high-dimensional and nonlinear problems . However, when applied to fourth-order biharmonic equations, PINNs still encounter critical bottlenecks: on the one hand, the direct computation of fourth-order derivatives is prone to vanishing gradients or numerical oscillations, significantly compromising accuracy; on the other hand, the presence of multiple boundary conditions and coupling constraints in such equations often leads to training imbalance under traditional static loss-weighting schemes, which can result in slow convergence or convergence to suboptimal local minima.

To address the aforementioned issues, this paper proposes a coupled PINNs method based on self-adaptive loss balancing (CLB-PINN). The core innovations of this method are twofold. Firstly, it adopts a dual-network coupled architecture. By introducing auxiliary variables, the original fourth-order equation is transformed into a second-order coupled system, which avoids the direct computation of high-order derivatives and thereby mitigates numerical errors. Secondly, it incorporates a Gaussian likelihood weighting module [8] to dynamically optimize the weight distribution across various loss terms. This enables the adaptive resolution of loss imbalance inherent in multi-objective training.

The remainder of this paper is structured as follows. Section 1 provides a brief introduction to the biharmonic equation

and physics-informed neural networks. Section 2 elaborates on the proposed CLB-PINN methodology. Section 3 validates the method through numerical experiments. Finally, Section 4 offers concluding remarks.

2. Methods

2.1 Review of physics-informed neural networks

We consider the biharmonic equation in the following form:

$$\begin{cases} \Delta^2 u = f \quad , \boldsymbol{x} \in \Omega, \\ u = g_1, , \dfrac{\partial^m u}{\partial \boldsymbol{n}^m} = g_2 \quad \boldsymbol{x} \in \partial\Omega. \end{cases} \tag{2.1}$$

This equation typically models the static deflection of an elastic plate under transverse lo ading. Here, Δ^2 deno tes the biharmonic operator, is the source term, g_1 and g_2 are prescribed functions, and $u(\boldsymbol{x})$ represents the unknown solution. The parameter m defines the boundary condition type: for $m=1$, the condition is referred to as the clamped boundary condition; for $m=2$, it is termed the simply-supported boundary condition.

A deep neural network $\hat{u}_\theta(\boldsymbol{x},\theta)$, parameterized by θ is first constructed. It takes spatial coordinates $\boldsymbol{x}$ from the domain Ω as input and outputs an approximation of the solution. For the biharmonic equation in (2.1), the corresponding loss function is defined as:

$$\mathcal{L}(\theta) = \frac{w_r}{N_r}\sum_{i=1}^{N_r}\left|\Delta\hat{u}_\theta(\boldsymbol{x}_r^i,\theta) - f\right|^2 + \frac{w_{dc}}{N_b}\sum_{i=1}^{N_b}\left|u_\theta(\boldsymbol{x}_b^i,\theta) - g_1(\boldsymbol{x}_b^i)\right|^2 + \frac{w_{nc}}{N_b}\sum_{i=1}^{N_b}\left|\frac{\partial^m \hat{u}_\theta}{\partial \boldsymbol{n}^m}(\boldsymbol{x}_b^i,\theta) - g_2(\boldsymbol{x}_b^i)\right|^2, \tag{2.2}$$

where w_r w_{dc} w_{nc} are the weights for each loss component. The sets $\left\{\boldsymbol{x}_r^i\right\}_{i=1}^{N_r}$, $\left\{\boldsymbol{x}_b^i\right\}_{i=1}^{N_b}$ represent the collections of interior and boundary training points, respectively. The parameters of the neural network model are iteratively updated by minimizing this composite loss function, yielding the approximate solution $\hat{u}_\theta(\boldsymbol{x},\theta)$ that converges to the unknown solution of the equation.

2.2 Biharmonic operator order-reduction technique and self-adaptive loss weighting

In standard PINNs, derivatives of all orders are computed via automatic differentiation. However, the numerical error tends to amplify exponentially with each increase in the derivative order, and the computational cost also rises substantially. These issues pose a major challenge when solving equations with high-order derivatives, such as the biharmonic equation. To address these issues, we first introduce an auxiliary variable $v = u(x)$. This allows the original fourth-order equation to be reformulated as a coupled system of two second-order partial differential equations:

$$\begin{cases} \Delta v = f(x,u,v), \\ v = \Delta u. \end{cases} \tag{2.3}$$

Consequently, while the form of the clamped boundary condition remains unchanged, the simply-supported boundary condition can be rewritten as:

$$u = g_1(x), v = g_2(x), x \in \partial\Omega. \tag{2.4}$$

The PINN architecture is then configured with two output neurons, enabling the simultaneous approximation of both u and v after training. Accordingly, the loss functions for problems with clamped and simply-supported boundary conditions are modified as follows, respectively:

$$\mathcal{L}(\theta)=\frac{w_r}{N_r}\sum_{i=1}^{N_r}\left|\Delta\hat{u}_\theta(\boldsymbol{x}_r^i,\theta)-v_\theta(\boldsymbol{x}_b^i,\theta)\right|^2+\frac{w_r}{N_r}\sum_{i=1}^{N_r}\left|v_\theta(\boldsymbol{x}_r^i,\theta)-f(\boldsymbol{x}_r^i,u_\theta,v_\theta)\right|^2+\frac{w_{dc}}{N_b}\sum_{i=1}^{N_b}\left|\hat{u}_\theta(\boldsymbol{x}_b^i,\theta)-g_1(\boldsymbol{x}_b^i)\right|^2+\frac{w_{nc}}{N_b}\sum_{i=1}^{N_b}\left|\frac{\partial\hat{u}_\theta}{\partial\boldsymbol{n}}(\boldsymbol{x}_b^i,\theta)-g_2(\boldsymbol{x}_b^i)\right|^2, \tag{2.5}$$

$$\mathcal{L}(\theta)=\frac{w_r}{N_r}\sum_{i=1}^{N_r}\left|\Delta\hat{u}_\theta(\boldsymbol{x}_r^i,\theta)-v_\theta(\boldsymbol{x}_b^i,\theta)\right|^2+\frac{w_r}{N_r}\sum_{i=1}^{N_r}\left|v_\theta(\boldsymbol{x}_r^i,\theta)-f(\boldsymbol{x}_r^i,u_\theta,v_\theta)\right|^2+\frac{w_{dc}}{N_b}\sum_{i=1}^{N_b}\left|\hat{u}_\theta(\boldsymbol{x}_b^i,\theta)-g_1(\boldsymbol{x}_b^i)\right|^2+\frac{w_{nc}}{N_b}\sum_{i=1}^{N_b}\left|v_\theta(\boldsymbol{x}_b^i,\theta)-g_2(\boldsymbol{x}_b^i)\right|^2. \tag{2.6}$$

Although the introduction of auxiliary variables resolves the challenges of approximating fourth-order derivatives and reduces computational cost, it simultaneously introduces new coupling dependencies among multiple loss terms. A fixed-weight loss scheme fails to adapt to this dynamic coupling, which can significantly undermine the advantages gained from the order reduction. Inspired by the adaptive loss balancing method proposed in [8], which updates adaptive weights based on maximum likelihood estimation to automate the allocation of weights across loss terms, we employ this strategy in our training process. The following numerical experiments will demonstrate the effectiveness of combining these two key techniques. The accuracy of our method is evaluated using the common L_2 error and maximum absolute error, defined as:

$$e_2=\frac{\sqrt{\sum_{j=1}^{N_{\text{data}}}\left|\hat{u}(x_j,t)-u(x_j,t)\right|^2}}{\sqrt{\sum_{j=1}^{N_{\text{data}}}\left|u(x_j,t)\right|^2}},\quad e_\infty=\max_{1\le j\le N_{\text{data}}}\left|\hat{u}(x_j,t)-u(x_j,t)\right|. \tag{2.7}$$

3. Results

3.1 Biharmonic Equation with Clamped Boundary Conditions

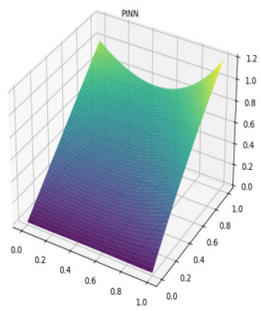
We first consider a two-dimensional biharmonic equation with clamped boundary conditions:

$$\begin{cases}\Delta^2 u=f, & (x,y)\in\Omega,\\ u=g_1,\ \dfrac{\partial u}{\partial n}=g_2, & (x,y)\in\partial\Omega,\end{cases} \tag{3.1}$$

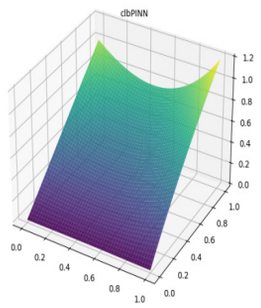
in which $\Omega=[0,1]^2$. The right-hand side of the equation and the boundary conditions are derived from the following exact solution: $u(x,y)=x^3\ln(1+y)+\dfrac{y}{1+x}$.

The equation (3.1) is solved using both the standard PINN framework and the proposed CLB-PINN method. Each neural network consists of 4 hidden layers with 32 neurons per layer, employing the hyperbolic tangent (tanh) activation function. The models are trained for 20,000 iterations using the Adam optimizer with an initial learning rate $\eta=0.001$. For training data, a set of 2,300 points is generated using Latin Hypercube Sampling (LHS), covering both the interior of the domain and its boundaries.

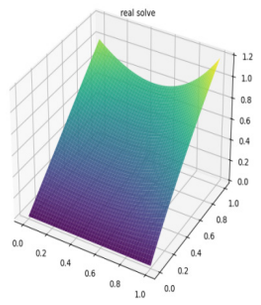
Figure 1 presents a comparative analysis of the results. As summarized in Table 1, it is evident that while the accuracy of the proposed CLB-PINN method remains within the same order of magnitude as the standard PINN in this case, its required computational time is significantly lower, demonstrating a substantially reduced cost.



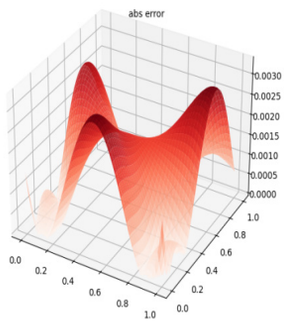
(a) PINN solution



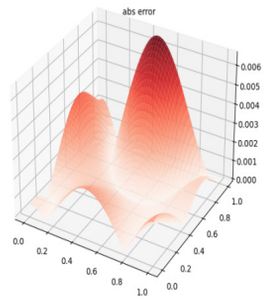
(b) CLB-PINN solution



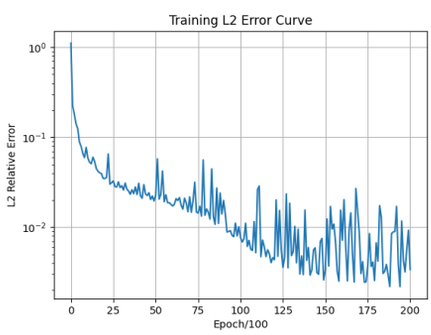
(c) Reference solution



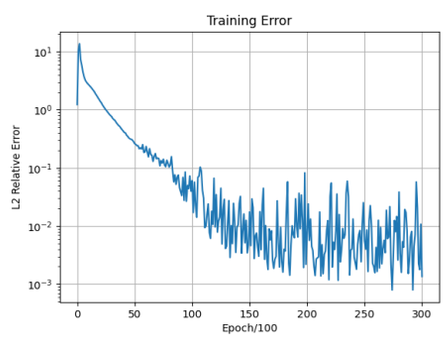
(d) Maximum absolute error of PINN



(e) Maximum absolute error of CLB-PINN



(f) L_2 relative error curve of PINN



(g) L_2 relative error curve of CLB-PINN

Figure 1. Comparison of solutions generated by different methods

Table 1. Comparison of time cost and solution accuracy

	Times	e_∞	e_2
PINN	2188s	3e-03	3.38e-03
CLB-PINN	1071s	6e-03	3.82e-03

3.2 Irregular star-shaped domain

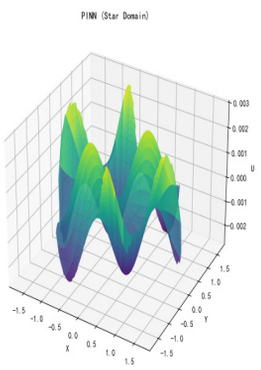
To demonstrate the robustness of our method on complex geometries, we next solve a two-dimensional biharmonic equation with simply-supported boundary conditions defined on an irregular star-shaped domain:

$$\begin{cases} \Delta^2 u + 18u = f, & (x,y) \in \Omega, \\ u = g_1, \ \Delta u = g_2, & (x,y) \in \partial\Omega, \end{cases} \tag{3.2}$$

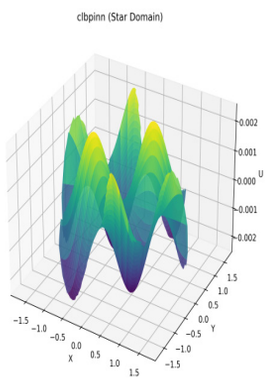
where the solution domain Ω is a star-shaped domain, described by the following parametric equations:

$$x = \rho(\theta)\cos\theta, y = \rho(\theta)\sin\theta, 0 \le \theta \le 2\pi, \ \ \rho(\theta) = \frac{(25 - 5\cos(6\theta - 3\pi))}{18} \tag{3.3}$$

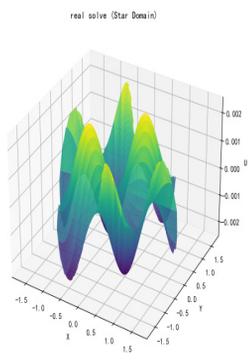
The right-hand side of the equation and boundary conditions are derived from the exact solution $u = \frac{\sin(\pi x)\sin(\pi y)}{4\pi^4 + 18}$. The equation (3.2) is solved using both the standard PINN framework and the proposed CLB-PINN method. The neural network for each model consists of 4 hidden layers with 64 neurons per layer, utilizing tanh activation function. The optimization is carried out using the Adam optimizer over 20,000 iterations, with an initial learning rate as specified. A total of 3,800 collocation points is sampled from both the interior and the boundaries of the domain for training. The comparative results are visualized in Figure 2. As clearly evidenced by the quantitative metrics in Table 2, the proposed method demonstrates superior performance to the standard PINN in terms of both solution accuracy and computational efficiency.



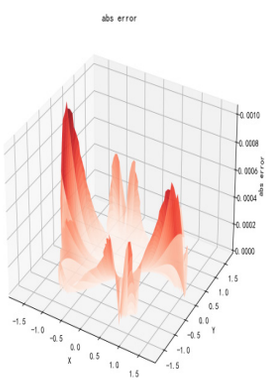
(a) PINN solution



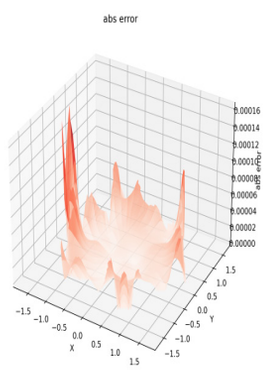
(b) CLB-PINN solution



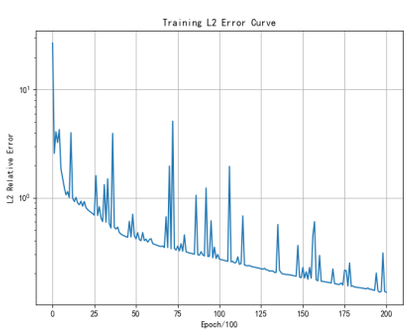
(c) Reference solution



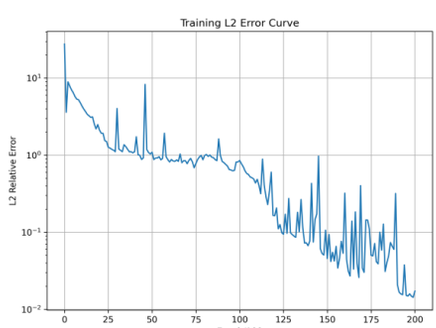
(d) Maximum absolute error of PINN



(e) Maximum absolute error of CLB-PINN



(f) L_2 relative error curve of PINN



(g) L_2 relative error curve of CLB-PINN

Figure 2. Comparison of solutions generated by different methods

Table 2. Comparison of time cost and solution accuracy

	Times	e_∞	e_2
PINN	3157s	1.04e-03	1.33e-01
CLB-PINN	515s	1.64e-04	1.71e-02

4. Conclusions

In this work, we have addressed key challenges in solving the biharmonic equation with traditional Physics-informed neural networks (PINNs)—notably, the inaccurate approximation of high-order derivatives, high computational cost, and difficulties in loss-function optimization. To overcome these limitations, we proposed a coupled PINNs method based on self-adaptive loss balancing (CLB-PINN). The main contributions of the method are twofold. Firstly, by introducing an auxiliary variable, the original fourth-order biharmonic equation is decomposed into a coupled system of two second-order partial differential equations. This reformulation avoids the direct computation of fourth-order derivatives, thereby significantly reducing numerical errors and computational expense. Secondly, an adaptive loss-weighting strategy is incorporated to dynamically update the weights of individual loss terms, effectively resolving the imbalance among multiple constraints during training. For future work, the CLB-PINN framework can be extended to higher-dimensional and nonlinear biharmonic equations, as well as to multi-physics coupled problems. Further improvements could also focus on designing more efficient network architectures and sampling strategies to enhance the method's applicability and performance in complex engineering scenarios.

References

[1] J. Hadamard, Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées. Mém. Sav. Étrang., 33 (1907) 1–128.

[2] G. Chen, Z. L. Li, P. Lin, A fast finite difference method for biharmonic equations on irregular domains and its application to an incompressible Stokes flow, Adv. Comput. Math. 29 (2) (2008) 113-133.

[3] J. N. Flavin, Convexity considerations for the biharmonic equation in plane polars with applications to elasticity, Q. J. Mech. Appl. Math. 45 (4) (1992) 555-566.

[4] M. M. Gupta, A finite difference method for the biharmonic equation with applications to plane stress and plate bending problems, Comput. Struct. 9 (2) (1978) 133–137.

[5] N. J. Altiero, D. L. Sikarskie, A boundary integral method applied to plates of arbitrary plane form. Comput. Struct. 9 (2) (1978) 163–168.

[6] B. P. Lamichhane, A stabilized mixed finite element method for the biharmonic equation based on biorthogonal systems, J. Comput. Appl. Math. 235 (2011) 5188–5197.

[7] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686-707.

[8] Z. Z. Xiang, W. Peng, X. Liu, W. Yao, Self-adaptive loss balanced physics-informed neural networks. Neurocomput. 496 (2022) 11–34.