J. Appl. Comput. Mech., 12(3) (2026) 1047-1068
DOI: 10.22055/jacm.2025.48578.5339

ISSN: 2383-4536
jacm.scu.ac.ir

Shahid Chamran University of Ahvaz

Journal of Applied and Computational Mechanics

Research Paper

A Mesh-Free Physics-Informed Neural Network Framework for Solving Plane Problems without Locking Issues

Zhipeng Lin[1,2], Ke Du[1,2], Zepeng Li[1,2]

[1] Key Laboratory of Earthquake Engineering and Engineering Vibration, Institute of Engineering Mechanics, China Earthquake Administration, Harbin 150080, China, Email: duke@iem.ac.cn (K.D.)

[2] Key Laboratory of Earthquake Disaster Mitigation, Ministry of Emergency Management, Harbin 150080, China, Email: 1208925405@qq.com (Zh.L.)

Received July 11 2025; Revised September 03 2025; Accepted for publication December 11 2025.
✉ Corresponding author: K. Du (duke@iem.ac.cn)

Abstract. This study proposes a plane problem solving method based on Physics-Informed Neural Network (PINN) for dealing with plane stress and strain problems, effectively avoiding the common shear locking and volume locking phenomena in traditional finite element methods, and exploring their underlying mechanisms. For the problem of plane stress, a PINN model based on strong form partial differential equations was constructed by randomly generating interior and boundary points within the solution domain, embedding the stress equilibrium equations and constitutive relations into the loss function. By minimizing the loss function, the network can automatically learn stress and displacement fields that satisfy the governing equations and boundary conditions. Numerical experiments show that this method can accurately predict stress distribution, effectively avoid shear locking, and maintain high accuracy even in thin plate structures. Further extension of the method to plane strain problems also successfully solved volume locking. This study also systematically compares two loss function construction methods based on strong form and energy principle (weak form). The results show that PINN, with its continuous and meshless solving characteristics, can fundamentally avoid the locking problem caused by low order element discretization, and both forms have high accuracy. This study validates the effectiveness and superiority of PINN in planar problems, providing new ideas and theoretical basis for overcoming traditional finite element locking problems. This method does not require complex element construction or numerical integration, and combines computational efficiency and accuracy, with good engineering application prospects.

Keywords: Plane problems, Neural networks, Machine learning, Shear locking, Volumetric locking.

1. Introduction

Plane stress and plane strain problems are classical problems in solid mechanics, with wide applications in engineering fields such as aerospace, civil engineering, and mechanical manufacturing.

Research on plane problems not only aids in understanding the behavior of materials under stress but also plays a crucial role in the design and analysis of engineering structures. Traditional numerical methods, such as the Finite Element Method (FEM) and Finite Difference Method (FDM), have achieved significant success in solving these problems. However, with the continuous development of engineering fields and the growing demand, there is an increasing urgency for more efficient and precise solutions to plane problems.

Over the past few decades, extensive research has been conducted in academia on plane stress problems, leading to the proposal of various theoretical models and solution methods. In 2011, Xue et al. [1] conducted a finite element analysis of the elastic thin plate bending problem using triangular elements, discussing the plane stress state of the plate-shell structure and the stress state during bending, which laid the foundation for subsequent FEM simulations. In 2013, Dang et al. [2] performed finite element analysis of solid and plane units to investigate the seismic performance of self-resetting shear walls. Using FEM software, they conducted nonlinear finite element analysis of self-resetting prestressed prefabricated shear walls and performed low-cycle cyclic load tests. A comparison revealed that the FEM model could effectively simulate mechanical properties such as lateral bearing capacity and unloading stiffness, but it was difficult to capture out-of-plane deformation and stress states. In 2015, Cao et al. [3] aimed to address the shortcomings of FEM, such as poor accuracy and slow convergence in engineering, by investigating the stress solution of a two-dimensional linear elastic perforated plate. They derived a stress interpolation matrix in polar coordinates, suitable for solving thin plates with circular holes, and applied it to a new finite element sequence, thus obtaining a new solution method. However, this method is only applicable to solving two-dimensional elastic perforated plates under tension or bending, limiting its generality. Additionally, mesh discretization in FEM still requires consideration of mesh density. These FEM analysis methods have inherent limitations. To improve computational accuracy, mesh density must be increased, which sacrifices computational efficiency, while

Published online: December 26 2025

prioritizing efficiency sacrifices accuracy. Moreover, when solving plane stress problems, shear locking can occur due to mesh discretization, leading to biased solutions.

Similarly, various theoretical models and solution methods for plane strain problems have been proposed over the past few decades. In 2006, Zheng et al. [4] employed a finite element limit analysis method to analyze geotechnical engineering problems. This method combines the advantages of numerical analysis and classical limit analysis. It first conducts strain analysis in two dimensions and then extends to three-dimensional analysis, contributing to geotechnical engineering design methods. Between 2014 and 2021, a Lagrangian framework-based Particle Finite Element Method (PFEM) [5-10] emerged in academia, which has been widely applied in large deformation simulations in geotechnical engineering. This method has simpler numerical implementation and greater flexibility. However, methods using low-order elements are susceptible to volumetric locking when simulating nearly incompressible solids. In 2023, Wang et al. [11] extended and improved the continuous limit analysis method (SLA), coupling it with the general numerical software OPTUM. This method shows greater generality and computational stability in analyzing the interaction between structures such as pipelines and undrained soft clay under large deformation in plane strain conditions. However, the above-mentioned FEM analysis methods are still mesh-dependent and require the introduction of mixed or higher-order elements to alleviate volumetric locking, which leads to a decrease in computational efficiency.

To overcome the limitations of traditional methods, many new numerical techniques have been proposed and applied to solving plane problems in recent years, such as mesh-free methods, isogeometric analysis methods, and machine learning methods. These new methods have addressed various challenges faced by traditional approaches to varying degrees and have shown great development potential. The original theory of mesh-free methods was proposed by Nayroles et al. [12], who discovered that constructing parallel paths for mesh-free approximations involves using the moving least squares approximation. They were the first to apply this technique within the Galerkin method, which was later named the Diffuse Element Method (DEM). Subsequently, Belytschko et al. [13] improved this method and named it the Element-Free Galerkin (EFG) method. In recent years, mesh-free methods have gained significant attention in academia for solving plane stress and plane strain problems due to their advantages, such as avoiding mesh distortion, simplifying preprocessing, handling discontinuities flexibly, high accuracy, and ease of parallel computation. As a result, many mesh-free methods have been developed for solving plane problems. For example, the Modified Variable Method proposed by Cho et al. [14], and the method proposed by Noguchi et al. [15] to improve the order of basis functions and expand the influence domain. However, numerical calculations by Kanok-Nukulchai [16] indicated that increasing the order of basis functions and expanding the influence domain in mesh-free methods still result in some shear locking, which has not been completely eliminated. On the other hand, the Matching Approximate Field Method proposed by Donning et al. [17] enables accurate calculation of large displacements and stresses, while the Stable Coordinate Node Integration Method by Chen [18-19] demonstrates significant improvements in computational efficiency, stability, and convergence. It is evident that using mesh-free methods to avoid the drawbacks of FEM is feasible and superior in both theory and practice for solving plane problems.

At the same time, research on machine learning methods has been exceptionally active, with numerous revolutionary achievements in various fields based on machine learning computation. For example, image recognition technology [20] and cardiovascular blood flow modeling [21], among others. In the engineering field, machine learning has also shown great promise. For instance, Zhang et al. [22] used a data-driven method based on Physics-Guided Convolutional Neural Networks (PhyCNN) for structural seismic response modeling. Su et al. [23] proposed a Gaussian process machine learning method to meet the high accuracy and speed requirements for slope stability evaluation. Chen [24] proposed an infrared image segmentation and recognition-based thermal defect detection system to address the need for rapid thermal defect detection in cold regions, providing a novel approach for thermal defect detection in construction acceptance and energy efficiency testing. The success of these computations relies on extensive labeled data training. However, in certain complex fields of physics and engineering, difficulties in data acquisition challenge the accuracy of model predictions. Moreover, after years of development, the engineering field has accumulated a wealth of physical models and partial differential equations (PDEs). In recent years, the combination of machine learning methods with traditional physical laws to solve the physical models represented by PDEs has emerged as a potential breakthrough in engineering problems. In 2019, Raissi et al. [25] proposed the Physics-Informed Neural Network (PINN) method for solving partial differential equations (PDEs), demonstrating the solution process. By eliminating the dependence on experimental and model data, they reduced the solution space and successfully solved the PDEs. Wang et al. [26] constructed a neural network model to solve the Reynolds-averaged equations and successfully predicted the variations in fluid Reynolds stresses. Wang et al. [27] incorporated the Boltzmann transport equation into neural networks, effectively predicting the multiscale heat transport problem of phonons in diamonds. In the field of solid mechanics, physics-informed neural networks (PINNs) have introduced new perspectives for solving a wide range of problems. Li et al. [28] developed neural network frameworks with different loss functions to solve elastic plate problems and compared the performance of the two approaches. Bai et al. [29] improved the loss function to solve two-dimensional solid mechanics problems and further extended their method to three-dimensional cases, offering new directions for applying neural networks to solid mechanics. Wang et al. [30] introduced a novel energy-based loss function that significantly enhanced the accuracy of neural network solutions, demonstrating that energy-based formulations are particularly well suited for solid mechanics applications. The PINN method combines partial differential equations (PDEs) with machine learning by embedding the PDEs into the neural network's loss function. This allows the network to simultaneously satisfy both the PDEs and boundary conditions, enabling efficient solutions even with limited or no data. This neural network model not only ensures accuracy but also exhibits improved generalization ability. Furthermore, it eliminates the need for discretization, avoiding mesh generation for the structure. In essence, it is a mesh-free method, forming the foundation for using PINN to solve plane stress problems in a mesh-free manner.

Building on these developments, this work systematically investigates PINN-based approaches for plane stress and plane strain problems, with a particular focus on elucidating the mechanisms by which PINNs overcome the shear locking and volumetric locking issues inherent in conventional finite element methods (FEM). Unlike prior studies that mainly applied PINNs to specific problems, the novelty of this work lies in probing the underlying mechanisms and rigorously validating the universality of the method. We analyze and explain the intrinsic reason behind the locking-free nature of PINNs: the use of a globally continuous and differentiable neural network as a trial function, which directly approximates solutions in the continuous domain. The optimization process seeks a globally smooth solution that either satisfies governing physical laws throughout the domain (strong form) or minimizes the system's total potential energy (weak form). This optimization paradigm, rooted in continuous function spaces, fundamentally avoids the excessive constraints caused by element discretization and low-order interpolation in FEM—thereby providing the essential reason PINNs inherently circumvent locking. To substantiate this argument, we conduct a comparative study of two PINN formulations: one based on the strong form (Mean Squared Error (MSE) -based) and the other on the weak form (energy-based). This design aims to demonstrate that the ability to avoid locking is an inherent advantage of the mesh-free, continuous nature of PINNs, rather than a feature dependent on any particular form of physical information embedding. The consistently excellent results obtained from both formulations across plane stress and plane strain examples provide mutually reinforcing evidence. These findings collectively demonstrate, from multiple perspectives, the effectiveness and generality of PINNs in resolving locking phenomena, thereby offering stronger support for their reliability.

Overall, the results confirm that the proposed PINN approach not only eliminates the need for complex mesh generation and achieves efficient and accurate solutions to plane problems, but also, from a theoretical standpoint, clarifies and validates its intrinsic ability to avoid FEM locking. This provides a novel perspective and robust solution framework for numerical analysis in solid mechanics.

2. Shear Locking and Volumetric Locking Phenomena in Finite Element Method

2.1. Shear locking phenomenon

The essence of shear locking is that shear deformation occurs in elements that, in theory, should not experience shear deformation. This type of shear deformation is referred to as "accompanying shear" [28]. This phenomenon typically occurs under pure bending conditions, where the primary effect is an increase in the model's bending stiffness, resulting in smaller deformations. If the analytical solution to the governing differential equations can be obtained directly, shear locking will not occur. However, when using the Finite Element Method (FEM), shear locking arises due to mesh discretization, leading to errors in the results.

Figure 1(a) is a schematic diagram of a certain structure. As shown in Fig. 1(b), take a structure subjected to bending moments at both ends. As shown in Fig. 1(c), this model represents the plane section assumption in material mechanics. Under the action of a bending moment, the section undergoes pure bending deformation. The top surface experiences horizontal tensile stress, leading to elongation, while the bottom surface undergoes compressive stress, leading to contraction. The vertical lines remain unchanged in length and remain perpendicular to the top and bottom surfaces. At this point, no shear deformation occurs, and the shear stress is zero. The deformation of the section is purely bending, as per the theoretical model. As shown in Fig. 1(d), this model represents the deformation diagram of the section using the Finite Element Method (FEM). It can be observed that the top surface undergoes tensile deformation, while the bottom surface experiences compressive deformation. However, the vertical lines either elongate or shorten, indicating shear deformation. This consumes some of the deformation energy, leading to overly stiff elements and reduced bending deformation, which deviates from the theoretical calculation.

The Finite Element Method (FEM) can effectively mitigate shear locking by refining the mesh. Increasing the mesh density, especially in regions where shear locking is likely to occur, enhances the accuracy and stability of the model, thereby reducing the occurrence of shear locking. However, this leads to an increase in the number of element nodes, significantly raising the computational workload. Alternatively, using reduced integration methods, which reduce the number of integration points per element, can lower the stiffness in numerical calculations, thereby alleviating the occurrence of shear locking. However, if the mesh size is inappropriate, significant errors may still occur in the results, which is related to the "hourglass" phenomenon in FEM. Using non-conforming elements is also a method to alleviate shear locking; however, when the element shapes are highly distorted, substantial errors in the results may occur.

2.2. Volume locking phenomenon

Volumetric locking is a common numerical issue in finite element methods, mainly occurring when the material is nearly incompressible or almost incompressible, such as rubber materials, plastic deformation, or large deformation analysis. When the Poisson's ratio of the material approaches 0.5, the material exhibits nearly incompressible characteristics, where the volumetric strain should theoretically tend to zero. However, due to the limitations of the element interpolation functions in FEM, the element may fail to accurately describe this volume invariance, leading to an excessive constraint on the volumetric strain, which results in the phenomenon of volumetric locking. The essence of volumetric locking is that, although the element should theoretically maintain a constant volume, non-physical volumetric strains occur, causing the model stiffness to become too large and the deformation results to become too small, leading to a significant deviation between the computational results and the actual situation.

As shown in Fig. 2(a), a 1/4 circular ring element is used to compare volumetric locking, applying a certain load. To better observe and compare the results, Abaqus modeling software is used to simulate two cases: one with volumetric locking and one without. The vertical displacement at the structural mutation position is observed. As shown in Fig. 2(b), this model is the result of an FEM simulation using full integration, which leads to the appearance of volumetric locking. When volumetric locking occurs, the volumetric strain of the element is excessively constrained, causing abnormal strain distributions inside the element. At this point, the element's deformation is constrained by non-physical forces, leading to an overly stiff model. Additionally, the model's deformation results are too small, deviating significantly from the theoretical or experimental results. As shown in Fig. 2(c), this model uses selective reduced integration for the FEM simulation, which avoids the volumetric locking phenomenon. When volumetric locking is effectively prevented, the strain distribution within the elements is uniform, and the deformation results match closely with the theoretical or experimental results.

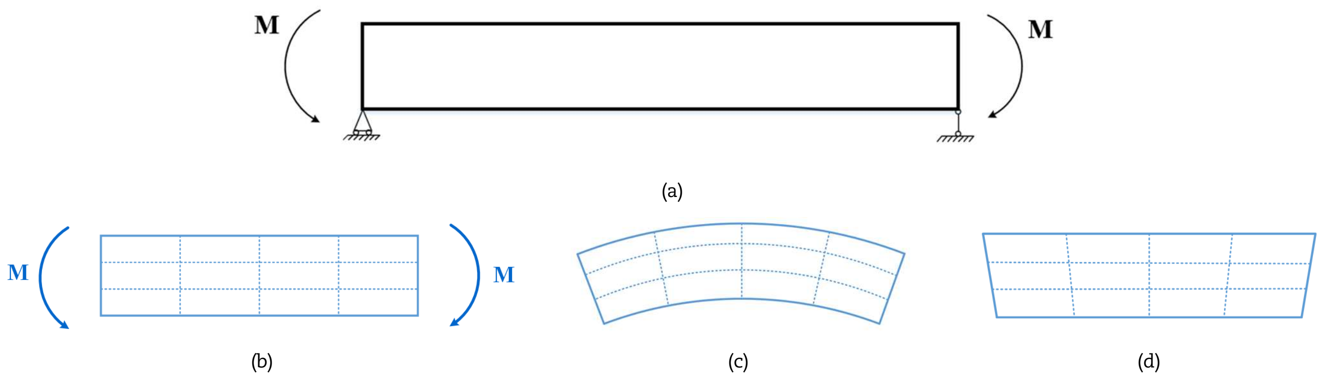


Fig. 1. Comparison of assumed deformation of plane section and finite element mesh generation method under bending moment action: (a) Structural schematic diagram, (b) Bending behavior of the structure, (c) Deformed shape under the plane section assumption, (d) Deformation profile of the FEM.

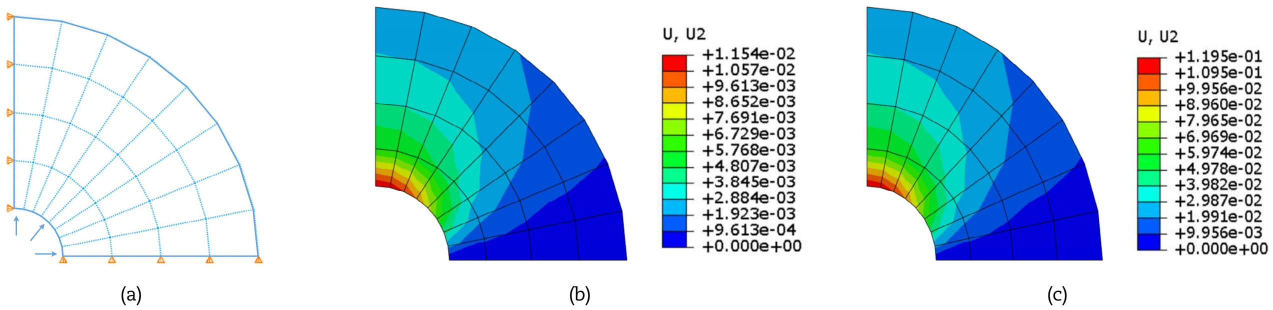


Fig. 2. Comparison of strain conditions with and without volumetric locking, (a) Structural loading representation, (b) FEM simulation with volumetric locking, (c) FEM simulation without volumetric locking.

Finite element methods can also improve volumetric locking by using higher-order elements, mixed elements, or hybrid elements to improve the interpolation functions' accuracy, thus reducing the occurrence of volumetric locking. However, these elements significantly increase the overall computational cost and make grid generation more complex. Additionally, using reduced or selective reduced integration methods to decrease the number of integration points in the element can lower its stiffness and alleviate the occurrence of volumetric locking. However, this method may introduce hourglass modes, which require extra control and are more complex to implement. Moreover, local mesh refinement can alleviate the volumetric locking phenomenon to a certain extent, but its effect is limited, and it cannot fundamentally solve the problem. A combination of other methods is needed to significantly improve the issue.

This paper proposes a Physics-Informed Neural Network (PINN) approach for solving the mechanical response in plane problems. Compared to traditional Finite Element Methods, this approach directly incorporates the physical laws into the neural network's loss function, avoiding the common shear locking and volumetric locking phenomena found in traditional numerical methods.

3. PINN Method for Solving Plane Stress Problems

3.1. Partial differential equations for plane stress problems

In a two-dimensional space, neglecting inertial effects, the plane stress problem has two key features: (1) All out-of-plane stress components vanish ($\sigma_z = \tau_{zy} = \tau_{zy} = 0$), leaving only in-plane stresses σ_x, σ_y, and τ_{xy} (note: τ_{xy} is uniformly denoted as σ_{xy} in this two-dimensional analysis). (2) Both stresses and strains are exclusively functions of x and y coordinates. Using the conservation of energy equation (force equilibrium equation) $\sigma_{ij,j} + f_i = 0$, the equilibrium equations in the x and y directions can be simplified into two separate equations. The equilibrium equation in the x direction is:

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + f_x = 0 \tag{1}$$

The equilibrium equation in the y direction is:

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + f_y = 0 \tag{2}$$

These two equilibrium equations together form the plane stress equilibrium equations. Here, σ_{xx} and σ_{yy} represent the normal stress components in the x and y directions, respectively, while $\sigma_{xy} = \sigma_{yx}$ is the shear stress component, and f_x and f_y are the body force components in the x and y directions, respectively.

At the same time, the plane stress problem must also account for displacements in the x and y directions, so the strain tensor needs to be specifically simplified for the x and y directions to obtain the corresponding equations. The normal strain component in the x direction is:

$$\varepsilon_{xx} = \frac{\partial u_x}{\partial x} \tag{3}$$

The normal strain component in the y direction:

$$\varepsilon_{yy} = \frac{\partial u_y}{\partial y} \tag{4}$$

The shear strain component in the x and y directions ($\varepsilon_{xy} = \varepsilon_{yx}$) is:

$$\varepsilon_{xy} = \frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y} \tag{5}$$

These three equations together form the geometric equations for the plane stress problem. Here, u_x represents the displacement in the x direction, and u_y represents the displacement in the y direction.

The plane stress problem must also satisfy the following constitutive equation:

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix} = \frac{E}{(1+\mu)(1-\mu)} \begin{pmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \dfrac{(1-\mu)}{2} \end{pmatrix} \begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{pmatrix} \tag{6}$$

where E is the elastic modulus for the plane stress problem, and μ is the Poisson's ratio for the plane stress problem.

The governing equations for the plane stress problem are formed by the equilibrium equations, geometric equations, and constitutive equations. In addition, when solving the plane stress problem, it is necessary to consider the boundary conditions of the elements. The boundary conditions include both stress boundary conditions and displacement boundary conditions.

The stress boundary conditions are:

$$\begin{cases} \left(l_1\sigma_{xx}+l_2\sigma_{xy}\right)_s=\overline{f_x} \\ \left(l_1\sigma_{xy}+l_2\sigma_{yy}\right)_s=\overline{f_y} \end{cases} \qquad (S=S_\sigma) \tag{7}$$

The displacement boundary conditions are:

$$\begin{cases} (u)_s=\bar{u} \\ (v)_s=\bar{v} \end{cases} \qquad (S=S_u) \tag{8}$$

where l_1 and l_2 represent the cosine values of the stress direction and the x and y axis directions, respectively. S, S_σ, and S_u represent all boundary surfaces of the structure, given the boundaries of stress and displacement. $\bar{f}_x$ and $\bar{f}_y$ represent the surface forces in the x and y directions, respectively, and $\bar{u}$ and $\bar{v}$ represent the displacements in the x and y directions, respectively.

These partial differential equations will endow the neural network model with physical meaning. The boundary conditions will be incorporated into the loss function, and by minimizing the loss function, the plane stress problem can be solved.

3.2. PINN model based on plane stress equations

3.2.1. *Architecture of the strong-form PINN model*

The architecture of the strong-form PINN model employed in this study is illustrated in Fig. 3.

Below is the description of the steps for establishing the PINN model:

1. The neural network used in this paper consists of an input layer, hidden layers, and an output layer. The network is a fully connected neural network $NN(x, y; \theta)$, where the input layer contains 2 neurons, representing the coordinates (x, y) of the points in the plane stress problem. After passing through $N\times M$ hidden layers (where N is the number of hidden layers and M is the number of neurons in each hidden layer), the output layer consists of 5 neurons. These neurons represent the solution to the plane stress problem, including the displacement along the x direction u_x, the displacement along the y direction u_y, the normal stress in the x direction σ_{xx}, the normal stress in the y direction σ_{yy}, and the shear stress σ_{xy}. This mapping can be mathematically expressed as:

$$NN(x,y)=\left(W^{L+1}\alpha_{L+1}\left(\ldots\alpha_2\left(W^2\alpha_1\left(W^1X^0+b^1\right)+b^2\right)\right)+b^{L+1}\right) \tag{9}$$

where the parameters W_i and b_i represent the weight matrix and bias vector for each layer of neurons, and α_i denotes the nonlinear activation function for each layer.

2. Random interior and boundary collocation points are generated within the solution domain [X, Y]. The output from the fully connected neural network $NN(x, y)$ automatically computes the multi-order gradients between points through automatic differentiation. Interior and boundary collocation points (also referred to as "collocation points") are randomly generated within the region [0, X]×[0, Y], allowing for the computation of their corresponding $\partial\sigma_{xx}/\partial x$, $\partial\sigma_{xy}/\partial x$, $\partial\sigma_{yy}/\partial y$, $\partial\sigma_{xy}/\partial y$, $\partial u_x/\partial x$, $\partial u_x/\partial y$, $\partial u_y/\partial x$, $\partial u_y/\partial y$, which are then incorporated into the model's loss function.

3. The PINN method solves the plane stress problem by training the neural network model to minimize the loss function of the points. Generally, the loss function of the PINN method consists of three parts: the error of the partial differential equation for the interior collocation points, the error of the boundary conditions for the boundary collocation points, and the error of the initial conditions. Since the plane stress problem is a static problem and does not consider initial conditions, this model only includes the errors from the partial differential equations and boundary conditions. This paper uses the Mean Squared Error (MSE) to measure the loss function, and the total loss function can be expressed as:

$$MSE=\varphi_{PDE}MSE_{PDE}+\varphi_{BC}MSE_{BC} \tag{10}$$

$$MSE_{PDE}=MSE_P+MSE_G+MSE_C \tag{11}$$

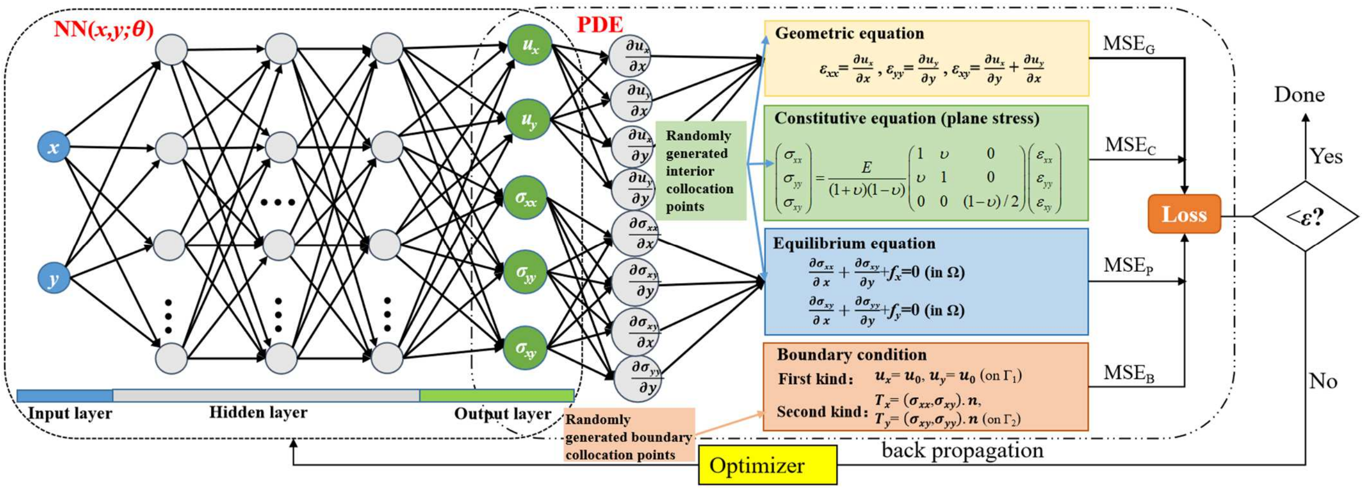


Fig. 3. Framework diagram of the strong-form PINN solution network.

Here, MSE_{PDE} represents the partial differential equation error, MSE_{BC} denotes the boundary condition error, while MES_P, MSE_G, and MSE_C correspond to the errors of equilibrium equations, geometric equations, and constitutive equations, respectively. φ_{PDE} and φ_{BC} are the weights in the loss function. There has been considerable research on the adaptive adjustment of these weights [32]. To simplify the training process, this study uses a pre-training phase of 5000 iterations before the main neural network training to automatically calculate the weights for each part. These weights will then be used in the formal training process to optimize the loss. Based on previous experience and pre-training tests, in this example, the weights are set as $\varphi_{PDE} = 1/E$ and $\varphi_{BC} = 1$, which can improve the training accuracy and speed. The specific expression for the PDE error is:

$$\begin{aligned}MSE_{PDE} = &\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\frac{\partial\sigma_{xx}}{\partial x}+\frac{\partial\sigma_{xy}}{\partial y}+f_x\right|^2+\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\frac{\partial\sigma_{xy}}{\partial x}+\frac{\partial\sigma_{yy}}{\partial y}+f_y\right|^2\\&+\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\varepsilon_{xx}-\frac{\partial u_x}{\partial x}\right|^2+\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\varepsilon_{yy}-\frac{\partial u_y}{\partial y}\right|^2\\&+\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\varepsilon_{xy}-\frac{\partial u_y}{\partial x}-\frac{\partial u_x}{\partial y}\right|^2+\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\sigma_{xy}-\frac{E}{2(1+\mu)}\varepsilon_{xy}\right|^2\\&+\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\sigma_{xx}-\frac{E}{(1+\mu)(1-\mu)}(\varepsilon_{xx}+\mu\varepsilon_{yy})\right|^2\\&+\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\sigma_{yy}-\frac{E}{(1+\mu)(1-\mu)}(\mu\varepsilon_{xx}+\varepsilon_{yy})\right|^2\end{aligned} \tag{12}$$

where N_p represents the number of random interior collocation points.

The strain components ε_{xx}, ε_{yy}, and ε_{xy} are directly substituted from Eqs. (3)-(5), ensuring that the PDE loss incorporates the equilibrium equations, geometric equations, and constitutive equations, thereby providing physical constraints for neural network training.

The specific expression for the boundary condition error is:

$$MSE_{BC} = \frac{1}{N_b}\sum_{i=1}^{N_b}\left|u_i-\overline{u_i}\right|^2+\frac{1}{N_b}\sum_{i=1}^{N_b}\left|\sigma_i-\overline{\sigma_i}\right|^2 \tag{13}$$

where N_b denotes the number of random boundary collocation points.

4. The physics-informed loss function continues to decrease as the model trains, and the training stops once the convergence conditions are met. If not, backpropagation is used to update the neural network parameters θ from Step 1, further reducing the loss. After training concludes, the displacement and stress values for the plane stress problem are output. Additionally, since this method does not involve mesh discretization and only optimizes the loss function for random interior collocation points and random boundary collocation points, it is inherently mesh-free and will not suffer from shear locking issues.

3.2.2. Energy-based loss function

To enhance the completeness and generalizability of the PINN method, this study investigates the use of an energy-based loss function (weak form) and compares it with the strong-form Mean Squared Error (MSE) -based loss function.

The energy-based loss function trains the neural network by minimizing the energy functional of the partial differential equations, where t denotes the traction load:

$$I = \int_\Omega \sigma\cdot\varepsilon dV - \int_{\partial\Omega} t\cdot u dA \tag{14}$$

Unlike hard-constrained PINNs, the energy-based loss function does not explicitly embed the balance equations and boundary terms into the loss function. Instead, the boundary conditions are naturally classified into Dirichlet and Neumann types, with the energy functional inherently accounting for the Neumann boundary conditions.

Consequently, the training loss function can be formulated by combining the energy functional with the boundary conditions:

$$L_{total} = L_{dbc} + L_{pde} \tag{15}$$

$$L_{dbc} = \frac{1}{N_{dbc}}\sum_{i=1}^{N_{dbc}}(u-u_{bc})^2 \tag{16}$$

$$L_{pde} = \sum_{i=1}^{N_{pde}} V_i\cdot\varepsilon\cdot(\mathbb{C}:\varepsilon) - \sum_{i=1}^{N_{nbc}} A_i\cdot t\cdot u \tag{17}$$

In these equations, V_i and A_i denote the control volume and surface area associated with each collocation point, respectively. N_{dbc}, N_{nbc}, and N_{pde} represent the number of collocation points subject to Dirichlet boundary conditions, Neumann boundary conditions, and located at integration points, respectively. It is worth noting that the boundary-condition loss term explicitly enforces only the Dirichlet conditions, since the Neumann conditions are naturally embedded in the numerical flux or weak form of Eq. (14).

By minimizing L_{total}, the predicted displacement field u naturally satisfies the principle of minimum potential energy, which corresponds to the weak form of the equilibrium equation. The framework based on the energy-based loss function is similar to that of the strong form Mean Squared Error (MSE) -based loss function, differing only in the internal loss term. In the subsequent analysis of plane strain problems, only the constitutive equations need to be modified accordingly. For a detailed discussion on the energy-based loss function, readers are referred to Li et al. [33] and Samaniego et al. [34].

3.2.3. *Mechanism of avoiding locking phenomena*

As discussed in Section 2, the mathematical essence of locking in finite element analysis arises from the conflict between overly strict constraints imposed during discretization (e.g., shear strain = 0 or volumetric strain = 0) and the inability of low-order interpolation functions to simultaneously satisfy both compatibility and incompressibility.

The PINN method fundamentally eliminates element-based discretization strategies, thereby avoiding this inherent contradiction. For strong-form PDE models, the loss function directly penalizes the residuals of the governing equations (equilibrium, constitutive, and geometric relations) and boundary conditions at the collocation points. The optimization of network parameters seeks a globally smooth and continuous function that simultaneously and flexibly satisfies all physical constraints across the domain. Without the limitations of low-order interpolation within elements, the neural network is not "locked" into spurious shear or volumetric strains, as occurs in linear finite elements. With its strong nonlinear representation capacity and global optimization nature, the network flexibly adjusts its functional form to spontaneously identify solutions that satisfy equilibrium, stress–strain, and displacement–strain relations. In this way, locking is naturally avoided, making PINN a function-space-based optimization paradigm.

For weak-form energy models, a more direct and rigorous theoretical basis for the non-locking property of PINNs is provided. The loss function is derived from the principle of minimum potential energy, directly minimizing the system's total potential energy. The key advantage of this formulation is the relaxation of constraints: the equilibrium equations need not be satisfied pointwise, but only in a weighted integral sense (i.e., the principle of virtual work). Within the energy minimization framework, provided that the neural network has sufficient expressive power and the optimization is effective, the solution naturally converges to the true solution, which is inherently free of locking. Thus, the weak form intrinsically avoids locking from a variational perspective.

The essential reason why PINNs avoid locking lies in their mesh-free nature and global continuous function approximation, rather than in a specific choice of loss function. Both strong and weak forms achieve this goal—through exact satisfaction of governing PDEs and through minimization of system energy, respectively. The strong form circumvents the pitfalls of low-order discretization via global optimization, whereas the weak form relaxes constraints at their variational root. Numerical examples presented later will demonstrate both approaches. This comparative design strongly supports the claim that avoiding locking is an intrinsic property of PINNs, thereby providing solid theoretical and experimental evidence for the core argument of this study.

3.3. Numerical example analysis

3.3.1. *Example 1*

In this section, Example 1 for the plane stress problem will be presented through the study of a two-dimensional plane stress pure bending beam problem to investigate the feasibility and effectiveness of the PINN method for solving plane stress problems. The structure of the problem is shown in Fig. 4(a). A simply supported beam with an elastic modulus E = 1000 Pa and Poisson's ratio μ = 0.3. The beam has a length of 1 m and width of 0.1 m. Bending moments M = 1/12 N·m are applied at both ends. To facilitate moment application, linearly distributed forces $F(y)$ are imposed on both sides of the beam, as illustrated in Fig. 4(b):

$$F(y) = 1000y \tag{18}$$

The stress boundary conditions are:

$$\begin{cases} \sigma_{xx}(0,y) = F(y) \\ \sigma_{xx}(1,y) = F(y) \\ \sigma_{xy}(0,y) = 0 \\ \sigma_{xy}(1,y) = 0 \\ \sigma_{xy}(x,0.05) = 0 \\ \sigma_{yy}(x,0.05) = 0 \end{cases} \tag{19}$$

The displacement boundary conditions are:

$$\begin{cases} u_x(0,-0.05) = 0 \\ u_y(0,-0.05) = 0 \\ u_y(1,-0.05) = 0 \end{cases} \tag{20}$$

Randomly, 10,000 random interior collocation points and 2,000 random boundary collocation points are selected to discretize the computational domain and boundary. This study adopts a neural network with 4 hidden layers, with 60 neurons in each hidden layer. The activation function used is the Tanh function.

The Finite Element Method (FEM) mentioned in this paper is modeled and simulated using Abaqus. The elements generated by the mesh are full-integration linear plane stress elements. As shown in the structural load diagram in Fig. 4(a), the Abaqus is used to fix both the horizontal and vertical displacements at the lower-left corner of the structure, and to fix the vertical displacement at the lower-right corner as the boundary conditions. Simultaneously, as shown in Fig. 4(b), a distributed load is applied to simulate the bending moments at both ends of the structure. Finally, the FEM results are output for comparison.

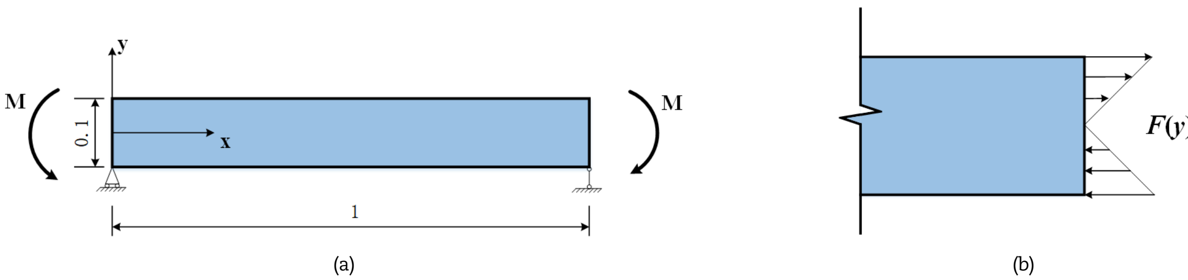


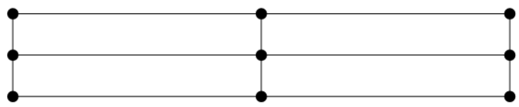
Fig. 4. Illustration of a pure bending beam and its equivalent load, (a) Structural loading representation, (b) Linear load distribution.

As shown in Fig. 5, the Finite Element Method (FEM) involves mesh discretization, where the numerical results at the sampling points are equivalent to the stress and displacement results of the mesh elements to which the sampling points belong. In contrast, the PINN method is a mesh-free approach, meaning that it does not require mesh generation. Instead, random sampling of collocation points within the plane is used to discretize the computational domain and boundary, which leads to a more efficient training process in the neural network. Fig. 6 shows the results of the FEM model with four elements. It can be observed that the FEM-generated spurious shear strain leads to shear locking. In the absence of nodes in the middle of the linear elements, pure bending cannot be simulated. The straight edges are unable to resist the bending load, and the excessive bending stiffness causes changes in the angle between the horizontal and vertical lines of the elements. The Abaqus FEM results were imported into Python and compared with the PINN results under the same legend. As shown in Fig. 7, the PINN method simulates the pure bending deformation more accurately than the FEM method. Moreover, the random selection of collocation points in PINN ensures that stress concentrations, which might be missed by the mesh in FEM, are properly captured.

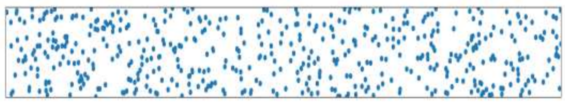
Refining the mesh can also alleviate shear locking to some extent. To further verify that the PINN method can avoid shear locking, the above example was discretized into 100,000 elements. As shown in Fig. 8, the FEM model with a fine mesh does not exhibit shear locking and simulates the deformation well. The Abaqus FEM results were again imported into Python and compared with the PINN results under the same legend. As shown in Fig. 9, the PINN results are nearly identical to those obtained using a refined finite element mesh, with only negligible error. This further confirms that PINNs can effectively prevent shear locking in plane stress problems.

To further assess the generality of PINNs under different physical constraints, we conducted parallel computations using both strong-form PDE constraints and weak-form energy (variational) models, followed by a systematic comparison. As illustrated in Fig. 10, the energy-based loss function accurately predicts the stress field distribution, with results in excellent agreement with the theoretical solution. This demonstrates that the energy-based and strong form Mean Squared Error (MSE) -based loss functions are equivalent in accuracy. In both cases, PINNs discretize the domain using random collocation points, inherently avoiding locking induced by mesh discretization in finite element methods. Through this comparison of strong and weak forms, the present case study provides compelling experimental evidence that the avoidance of locking is an inherent property of the mesh-free, continuous-solution paradigm of PINNs.

(a)



(b)



(c)

Fig. 5. Comparison of sampling analysis between FEM and PINN method, (a) Loading condition of the planar beam, (b) Finite element mesh representation, (c) Randomly sampled collocation points in the PINN framework.

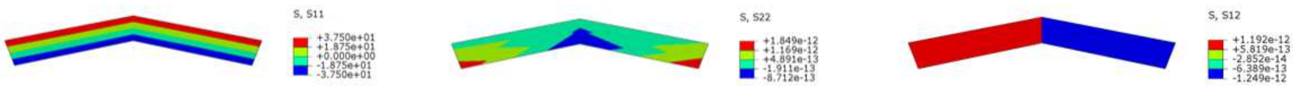
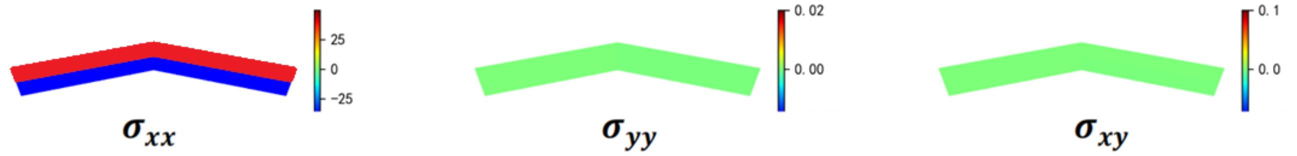
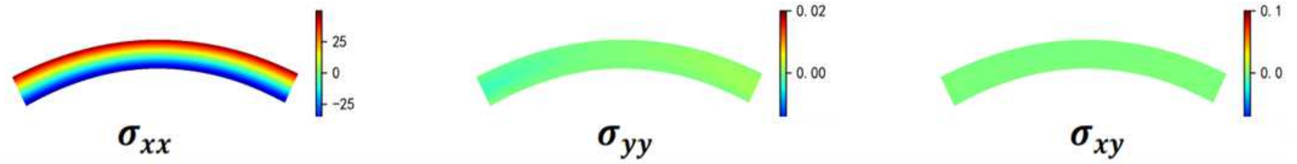


Fig. 6. Stress distribution from Abaqus results.



(a)



(b)

Fig. 7. Comparison of results between FEM and PINN method, (a) Stress distribution from FEM, (b) Stress distribution from PINN.

Fig. 8. Stress distribution from fine-mesh Abaqus results.

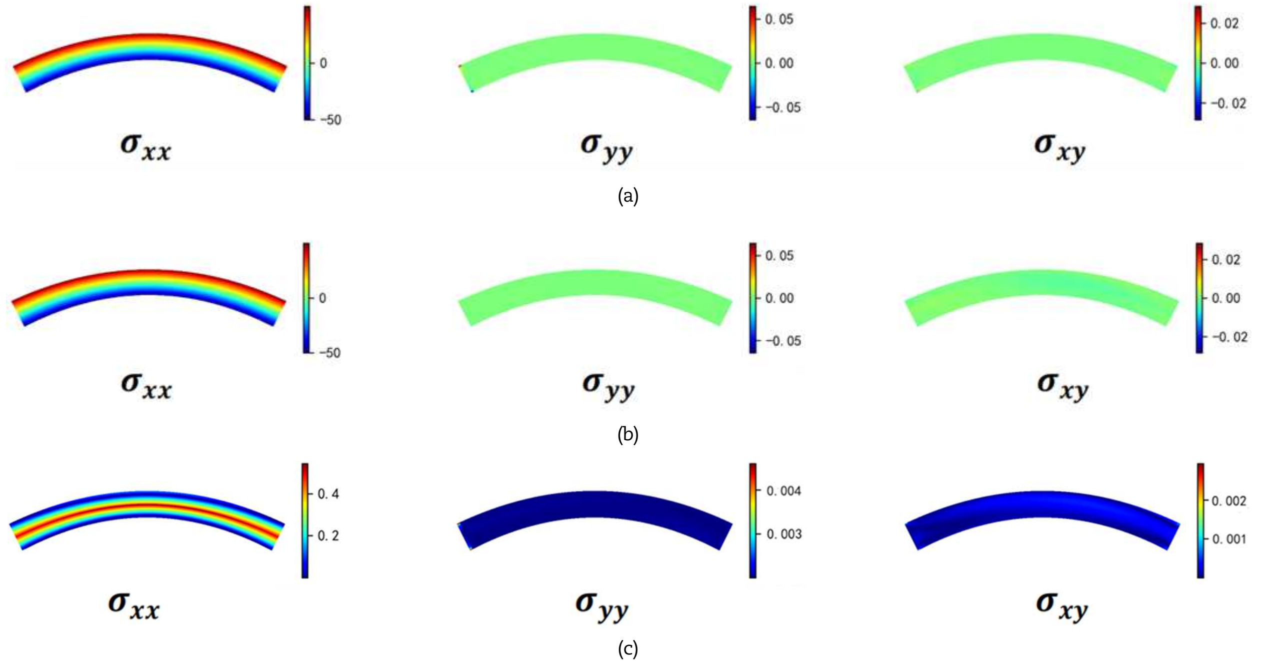


Fig. 9. Comparison between fine grid results and PINN method results, (a) Stress distribution from fine-mesh FEM, (b) Stress distribution from PINN, (c) Error distribution plot.

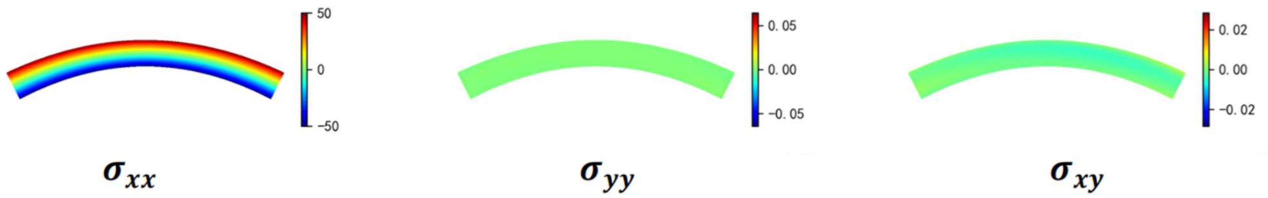


Fig. 10. Results obtained using the energy-based loss function.

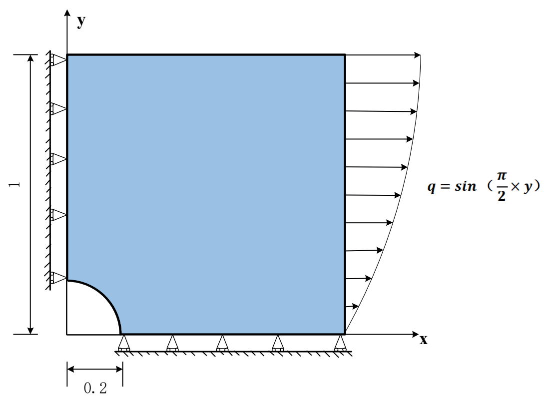


Fig. 11. Schematic of the defective plate structure.

3.3.2. *Example 2*

To verify the applicability of the proposed model under complex boundary and loading conditions, Example 2 of the plane stress problem in this study calculates the plane stress problem for a defective plate. The structure is shown in Fig. 11. The problem consists of a square plate with a quarter-circle defect at the lower-left corner. The elastic modulus E is 5 Pa, and the Poisson's ratio μ is 0.3. The side length of the square is 1 meter, the center of the circle is at (0, 0), and the radius is 0.2 meters. A distributed load of $q = \sin(y\times\pi/2)$ is applied on the right side of the square plate. The stress boundary conditions are:

$$\begin{cases} \sigma_{xx}(1,y) = \sin\left(\dfrac{\pi}{2}\times y\right) \\ \sigma_{xy}(x,0) = 0 \\ \sigma_{xy}(0,y) = 0 \\ \sigma_{xy}(1,y) = 0 \\ \sigma_{xy}(x,1) = 0 \\ \sigma_{yy}(x,1) = 0 \end{cases} \tag{21}$$

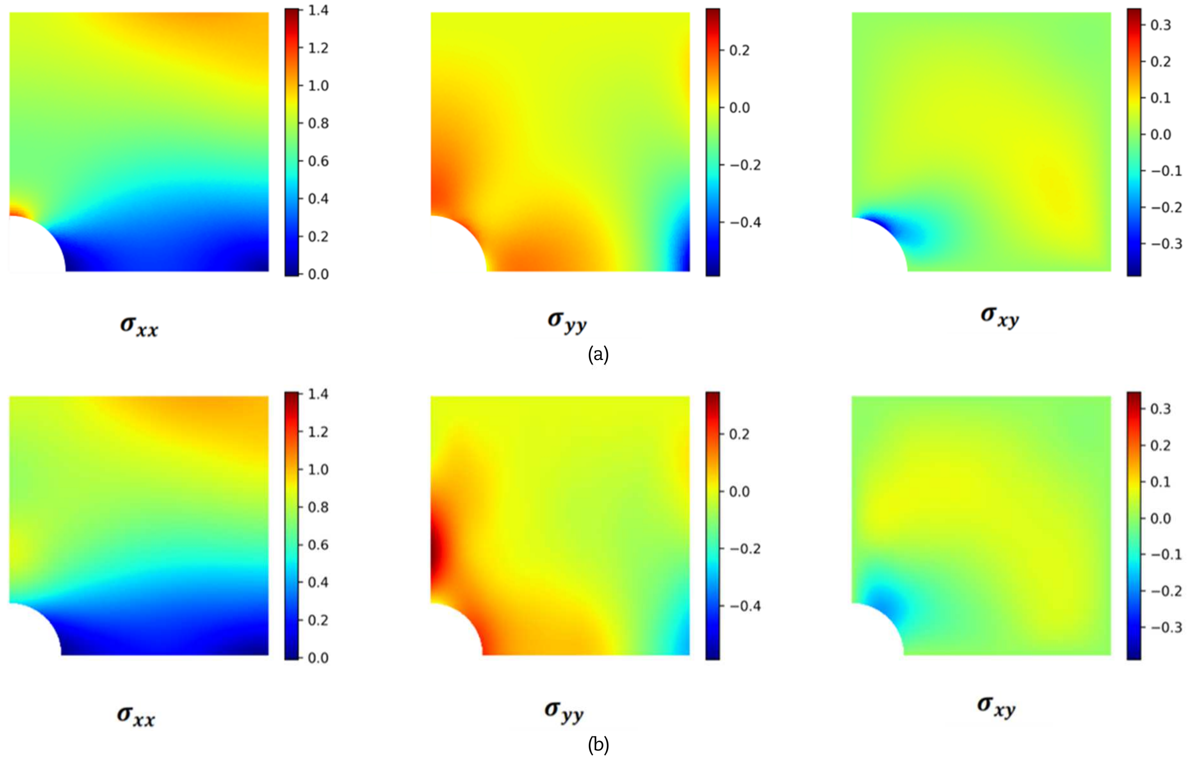


Fig. 12. Comparison of FEM and PINN method results for defective plates, (a) Stress distribution from FEM, (b) Stress distribution from PINN.

The arc boundary conditions are:

$$\begin{cases}\sigma_{xx}\times n_x+\sigma_{xy}\times n_y=0\\ \sigma_{xy}\times n_x+\sigma_{yy}\times n_y=0\end{cases} \tag{22}$$

where n_x and n_y represent the cosine values of the arc normal vector with respect to x and y axes. The displacement boundary conditions are:

$$\begin{cases}u_x(0,y)=0\\ u_y(x,0)=0\end{cases} \tag{23}$$

As shown in Fig. 12, the results obtained by the neural network under nonlinear boundary conditions and complex loading remain excellent. The computational results do not exhibit shear locking and show high accuracy.

When calculating Example 2, there is no need to rebuild the neural network structure. Only the boundary conditions in the neural network model need to be modified according to the model's boundary and loading conditions, allowing for the precise calculation of the structure's displacement and stress. In contrast, the Finite Element Method requires the model to be rebuilt, and the mesh to be redefined for each different example, making it significantly less efficient than the PINN method. It is evident that the PINN method is more advantageous than traditional FEM in solving plane stress problems.

3.4. Discussion on finite element mesh division

To investigate the effect of FEM mesh refinement on mitigating shear locking, this study uses Abaqus to discretize the structure into 4 elements, 1000 elements, 10000 elements, and 100000 elements. The vertical displacement at the center span is then compared, with results from the PINN method and the theoretical solution shown in Table 1.

In Example 1 from the previous section, the theoretical maximum displacement at the center span (with a default plate thickness of 1 in Abaqus) is calculated as:

From Table 1, it can be seen that FEM Model 1 exhibits a significant error compared to the other models, clearly showing shear locking. As the mesh density increases from FEM Model 2 to FEM Model 4, the vertical displacement at the center approaches the theoretical solution, but the computational time also gradually increases. It can be concluded that, disregarding computation speed, refining the structure's mesh can alleviate the shear locking issue.

Table 1. Vertical displacement at mid span under different grid divisions.

	Number of elements in width direction	Number of elements in length direction	Total number of elements	Mid-span vertical displacement value
FEM model 1	2	2	4	0.09403
FEM model 2	10	100	1000	0.124103
FEM model 3	100	100	10000	0.124796
FEM model 4	100	1000	100000	0.125062
Energy–based PINN model	/	/	/	0.125015
Mean Squared Error (MSE) - based PINN model	/	/	/	0.124982
PINN model	/	/	/	0.124982
Theoretical solution	/	/	/	0.125

However, both the energy–based PINN model and the Mean Squared Error (MSE) -based PINN model yield solutions that are highly consistent with the results obtained from fine-mesh finite element analysis, even without mesh discretization. This demonstrates that the PINN method can also effectively avoid shear locking while bypassing the laborious step of mesh generation, thereby providing further evidence that the avoidance of locking is an inherent property of the mesh-free, continuous solution paradigm of PINNs.

4. PINN Method for Solving Plane Strain Problems

4.1. Partial differential equations for plane strain problems

The partial differential equations for plane strain problems can be derived in a similar manner to those for plane stress problems. Similar to the plane stress problem, the plane strain problem assumes that the strain component in one direction (usually the z direction) is zero, i.e. $\varepsilon_z = 0$. However, unlike plane stress problems, the stress component σ_z in the z direction is non-zero in plane strain problems and is determined through constitutive relations with other stress components.

The governing equations for the plane strain problem are similar in form to those for the plane stress problem, particularly in the expressions for the equilibrium equations and the geometric equations. The main difference lies in the constitutive relations, which describe the relationship between stress and strain. The constitutive equation for the plane strain problem must account for the constraint in the z direction, so the partial differential equations for the plane strain problem are obtained by modifying the constitutive equations for the plane stress problem. The plane strain problem must satisfy the following constitutive equations:

$$\begin{pmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{xy}\end{pmatrix} = \frac{E}{(1+\mu)(1-2\mu)}\begin{pmatrix}2(1+\mu) & \mu & 0\\ \mu & 2(1+\mu) & 0\\ 0 & 0 & \dfrac{(1-2\mu)}{2}\end{pmatrix}\begin{pmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{xy}\end{pmatrix} \tag{24}$$

The parameter E represents the elastic modulus for the plane strain problem, and μ represents the Poisson's ratio for the plane strain problem. The governing equations for the plane strain problem are formed by the equilibrium equations, the geometric equations, and the constitutive relations. Additionally, when solving the plane strain problem, the boundary conditions of the element must also be considered. The boundary conditions for the plane strain problem are the same as those for the plane stress problem, so the boundary conditions for the plane stress problem discussed earlier can be directly used for the plane strain problem.

These partial differential equations provide physical meaning to the neural network model. The constraints are incorporated into the loss function, and by minimizing the loss function, the plane strain problem can be solved.

4.2. PINN model based on plane strain equations

The PINN model structure based on plane strain equations adopted in this paper is similar to that for plane stress problems, as shown in Fig. 13.

In this chapter, the same PINN framework as in Section 3 is employed, with the only modification being the replacement of the constitutive relation by its plane strain form. The key difference lies in the formulation of the constitutive relations: the PDE error term for the plane strain problem, MSE_{PDE}, is calculated using Eq. (25), where the strain-displacement relations are still given by Eqs. (3)–(5), but the constitutive relations are replaced with those in the plane strain form. This unified modeling framework allows the same network architecture to be adapted for both plane stress and plane strain problems by simply adjusting the constitutive relations, highlighting the versatility and flexibility of the PINN approach. Since a mesh-free method is employed, this model also avoids the volumetric locking phenomenon that typically arises in traditional Finite Element Methods (FEM) for plane strain problems:

$$\begin{aligned} MSE_{PDE} = {} & \frac{1}{N_p}\sum_{i=1}^{N_p}\left|\frac{\partial\sigma_{xx}}{\partial x}+\frac{\partial\sigma_{xy}}{\partial y}+f_x\right|^2+\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\frac{\partial\sigma_{xy}}{\partial x}+\frac{\partial\sigma_{yy}}{\partial y}+f_y\right|^2+\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\varepsilon_{xx}-\frac{\partial u_x}{\partial x}\right|^2+\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\varepsilon_{yy}-\frac{\partial u_y}{\partial y}\right|^2 \\ & +\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\varepsilon_{xy}-\frac{\partial u_y}{\partial x}-\frac{\partial u_x}{\partial y}\right|^2+\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\sigma_{xy}-\frac{E}{2(1+\mu)}\varepsilon_{xy}\right|^2+\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\sigma_{xx}-\frac{E}{(1+\mu)(1-2\mu)}\left[2(1+\mu)\varepsilon_{xx}+\mu\varepsilon_{yy}\right]\right|^2 \\ & +\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\sigma_{yy}-\frac{E}{(1+\mu)(1-\mu)}\left[2(1+\mu)\varepsilon_{yy}+\mu\varepsilon_{xx}\right]\right|^2 \end{aligned} \tag{25}$$

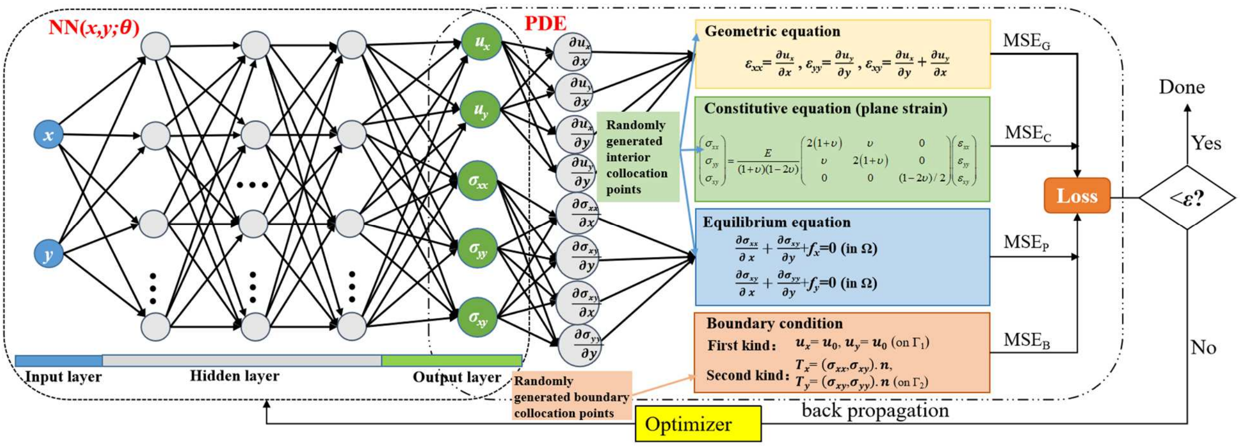


Fig. 13. Framework of the PINN-based solution network.

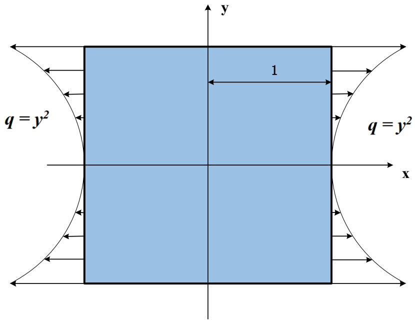


Fig. 14. Schematic representation of the square plate structure.

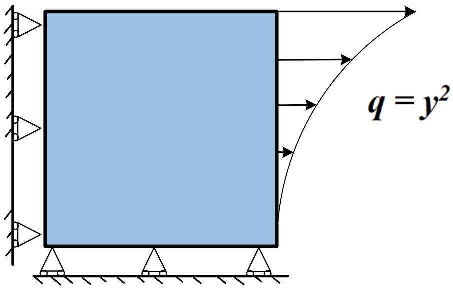


Fig. 15. Load representation of the simplified structure.

4.3. Numerical example analysis

4.3.1. *Example 3*

In Example 3 of the plane strain problem, a classic plane strain problem is used to study the feasibility and effectiveness of the PINN method for solving plane strain problems. The structure is shown in Fig. 14. It is a square plate of size 2×2, with nonlinear distributed load $q = y^2$ Pa applied on its left and right sides, while there is no load on the top and bottom sides. The material's elastic modulus E is 5 Pa, and the Poisson's ratio $\mu = 0.45$. Due to the geometric symmetry of the problem, it can be simplified into a quarter model, as shown in Fig. 15, for calculation.

Stress boundary conditions:

$$\begin{cases} \sigma_{xx}(1,y) = y^2 \\ \sigma_{xy}(1,y) = 0 \\ \sigma_{xy}(0,y) = 0 \\ \sigma_{xy}(x,0) = 0 \\ \sigma_{xy}(1,y) = 0 \\ \sigma_{yy}(x,1) = 0 \end{cases} \tag{26}$$

Displacement boundary conditions:

$$\begin{cases} u_x(0,y) = 0 \\ u_y(x,0) = 0 \end{cases} \tag{27}$$

A total of 10,000 random interior collocation points and 2,000 random boundary collocation points are used to discretize the computational domain and boundaries. The neural network model consists of 4 hidden layers, each containing 60 neurons, with the Tanh activation function applied.

In this example, Abaqus is used for modeling and simulation. The elements generated from the mesh are plane strain elements with global reduced integration and hourglass control. As shown in Fig. 15, the quarter model under load, Abaqus is used to fix the lateral displacement on the left side of the structure and the vertical displacement on the bottom side, representing the boundary conditions of the structure. Additionally, a nonlinear distributed load is applied to simulate the structural loading. Finally, the finite element results are output and compared with the PINN model results.

The mesh generation and collocation point sampling are the same as in Section 3.3.1 for the plane stress problem example, and will not be repeated here. The mesh density has a significant effect on the occurrence of volumetric locking. A coarse mesh results in larger element sizes and fewer degrees of freedom, making it difficult to accurately capture the incompressibility constraint of the material, which can lead to volumetric locking. On the other hand, a finer mesh reduces the likelihood of volumetric locking but cannot completely avoid it, and it also increases the overall computational cost. Furthermore, the choice of elements in the mesh has a considerable impact on the occurrence of volumetric locking. The PINN method, being a mesh-free approach, naturally adapts to complex geometries and avoids the volumetric locking phenomenon without the need for mesh generation. The results obtained through the PINN method outperform those from traditional finite element methods.

Figure 16 shows the displacement results obtained from finite element modeling of the case study using Abaqus software. To facilitate comparison with the PINN method, the Abaqus nodal data were imported into Python and synchronized with the PINN outputs. Figures 17 and 18 show the displacement variations calculated by the finite element method and the PINN method, respectively, allowing observation of the strain distribution induced by the applied loads. The finite element mesh consists of plane strain elements with global reduced integration and hourglass control, which prevents volumetric locking in the finite element solutions. Figure 19 presents the comparison of errors between the finite element method and the PINN method, providing a clear assessment of the accuracy of the PINN solutions. It can be observed that the results obtained by the PINN method are in good agreement with those from the finite element method, exhibiting minimal error, which demonstrates the effectiveness and feasibility of the PINN approach for plane strain problems. Moreover, the strain fields computed by the PINN method are smoother and continuous, allowing more accurate capture of local strain variations, thus mitigating numerical oscillations or discontinuities that may arise from finite element meshing. Additionally, the mesh-free nature of the PINN method greatly improves computational efficiency, eliminating the need for complex mesh generation and optimization inherent to finite element methods, and enhancing its applicability for problems with complex boundary conditions.

Similar to Case Study I, this example further examines the validity of the PINN method by comparing the results obtained from both the strong-form and weak-form modeling approaches. As shown in Fig. 20, the solutions obtained with the energy-based loss function are also of high accuracy. Although the formulations of the energy-based and Mean Squared Error (MSE) -based loss functions differ, both are implemented within the mesh-free collocation framework of PINNs. This further demonstrates that the mesh-free paradigm is the essential mechanism by which PINNs overcome locking.

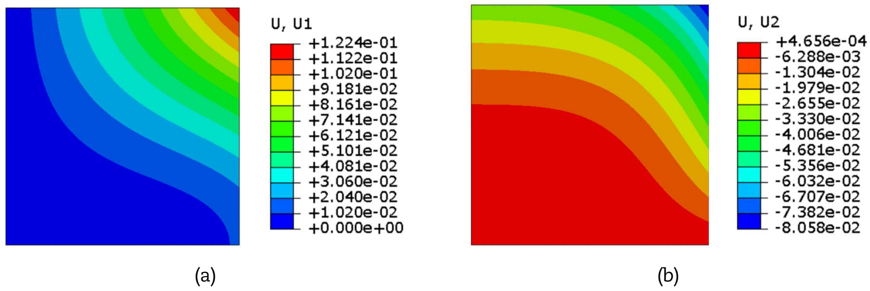


Fig. 16. Displacement results from Abaqus, (a) x direction displacement diagram, (b) y direction displacement diagram.

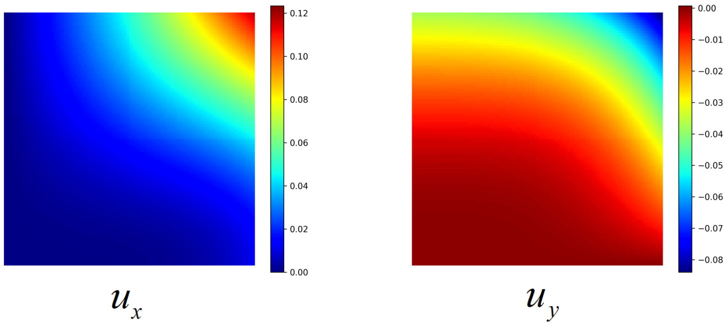


Fig. 17. Displacement field from FEM (Poisson's ratio = 0.45).

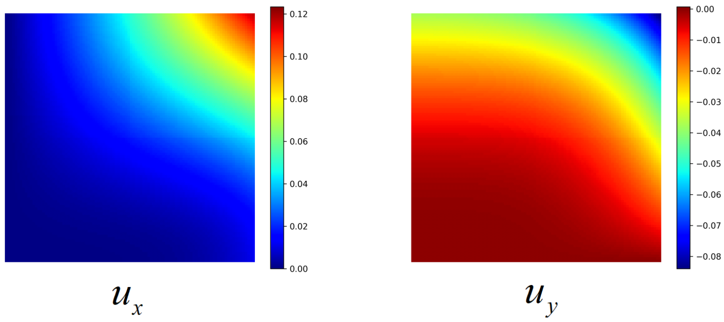


Fig. 18. Displacement field from PINN (Poisson's ratio = 0.45).

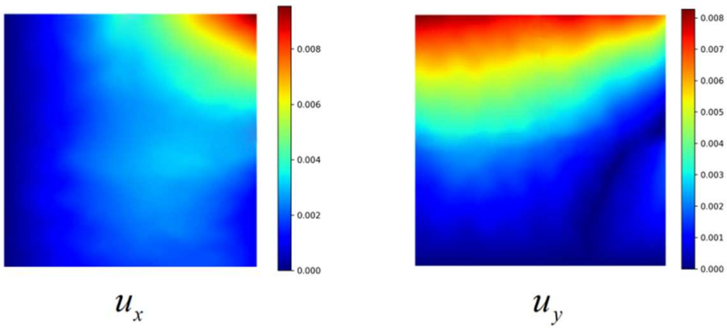


Fig. 19. Error comparison between the finite element method and the PINN method.

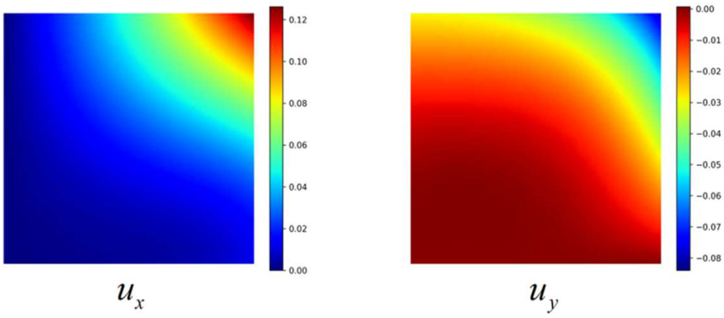
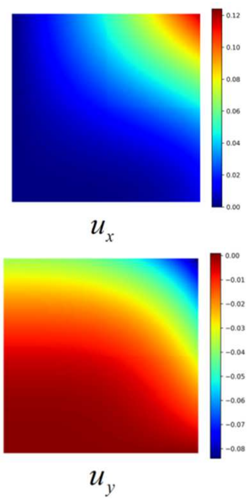
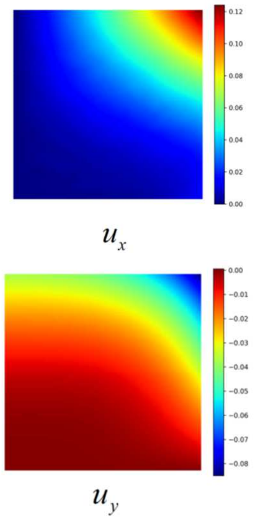
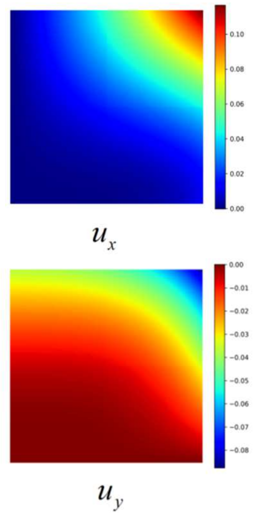
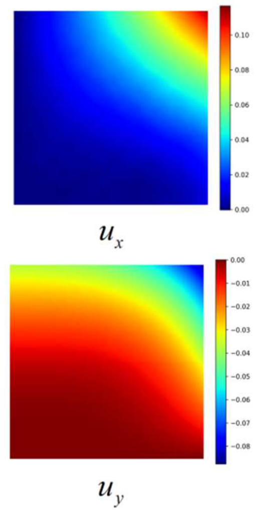
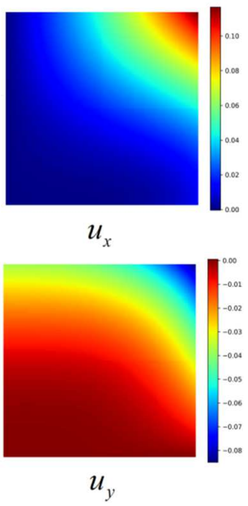
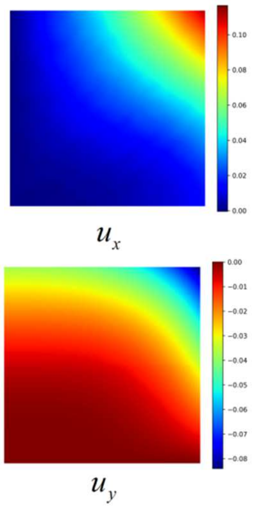
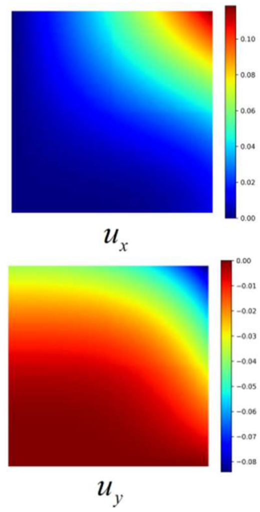
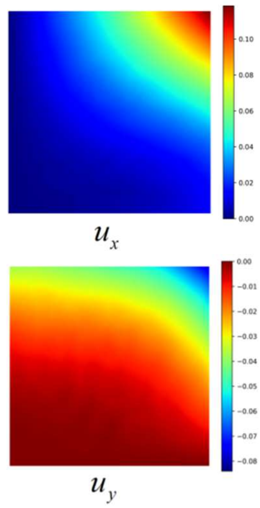


Fig. 20. Results obtained using the energy-based loss function.

Table 2. Comparison of results under different Poisson's ratios.

Poisson's Ratio μ	FEM Results	PINN Results	Poisson's Ratio μ	FEM Results	PINN Results
μ = 0.46			μ = 0.47		
μ = 0.48			μ = 0.49		

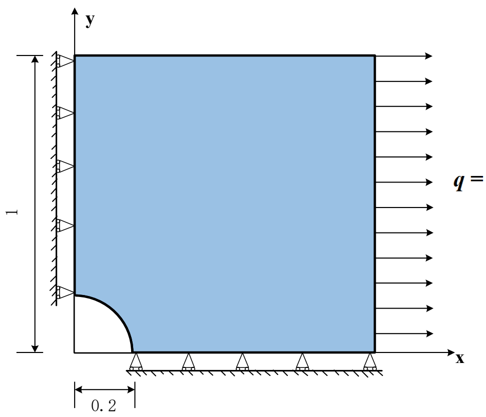


Fig. 21. Schematic of the defective plate structure.

To verify whether the PINN method is still applicable when the material's Poisson's ratio approaches 0.5, comparisons were made for Poisson's ratios of 0.46, 0.47, 0.48, and 0.49, as shown in Table 2. The results indicate that even as the Poisson's ratio increases, the PINN method maintains high accuracy, further confirming that the PINN approach can effectively avoid the occurrence of volumetric locking when solving plane strain problems.

4.3.2. Example 4

To verify the applicability of the neural network model under complex boundary conditions, Example 4 in this paper investigates a plane strain problem of a defective plate. The structure of Example 4 is shown in Fig. 21: a square plate with side length 1 m contains a quarter-circle defect (radius = 0.2 m centered at (0, 0)). The material has elastic modulus E = 5 Pa and Poisson's ratio μ = 0.45. A uniform distributed load q = 1 is applied on the right edge of the square plate.

Stress boundary conditions:

$$\begin{cases} \sigma_{xx}(1,y) = 1 \\ \sigma_{xy}(x,0) = 0 \\ \sigma_{xy}(0,y) = 0 \\ \sigma_{xy}(1,y) = 0 \\ \sigma_{xy}(x,1) = 0 \\ \sigma_{yy}(x,1) = 0 \end{cases} \tag{28}$$

Arc boundary conditions:

$$\begin{cases} \sigma_{xx} \times n_x + \sigma_{xy} \times n_y = 0 \\ \sigma_{xy} \times n_x + \sigma_{yy} \times n_y = 0 \end{cases} \tag{29}$$

where n_x and n_y represent the cosine values of the arc normal vector relative to the x and y axes.

Displacement boundary conditions:

$$\begin{cases} u_x(0,y) = 0 \\ u_y(x,0) = 0 \end{cases} \tag{30}$$

As shown in Figs. 22 and 23, a comparative analysis with the FEM using globally reduced integration plane strain elements with hourglass control clearly demonstrates the superior performance of the PINN method in the case of complex structural loading. In the simulation of nearly incompressible materials, the PINN method not only completely avoids the volumetric locking phenomenon commonly observed in traditional finite element methods, but also exhibits significant advantages in computational accuracy. This highlights the theoretical advantages of embedding physical information into the model.

To systematically evaluate the applicability of the neural network model for nearly incompressible structures under complex boundary conditions, a series of comparative experiments were conducted on the defect plate strain model for Poisson's ratios of 0.46, 0.47, 0.48, and 0.49, similar to the square plate strain model. The analysis results, as shown in Table 3, indicate that as the Poisson's ratio approaches the incompressible limit of 0.5, the PINN method consistently maintains stable computational accuracy. This characteristic thoroughly demonstrates the robustness of the proposed neural network architecture and adaptive weighting strategy under extreme conditions.

In terms of computational efficiency, the PINN method shows unique advantages. Compared to the square plate strain model, there is no need to reconstruct the network architecture when solving the defect plate strain model; only adjustments for new boundary conditions and loading situations are required to obtain an accurate solution. In contrast, traditional finite element methods require the geometric modeling, mesh generation, and finite element selection to be redone for each case. Particularly when dealing with nearly incompressible problems, finite element methods often need to use special hybrid or high-order strain elements to enhance solution accuracy and avoid volumetric locking, which further increases the computational complexity.

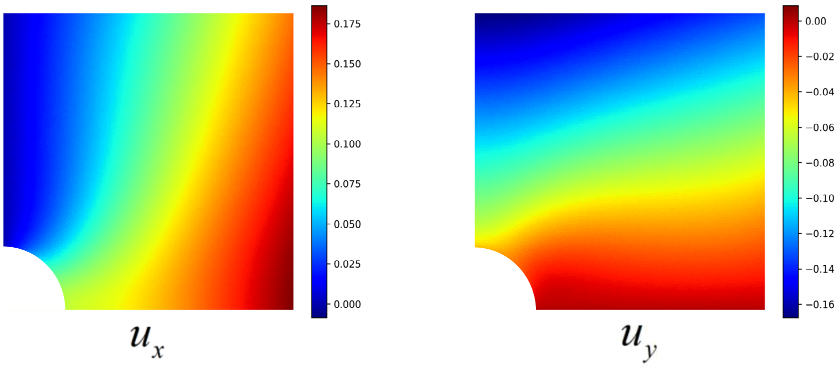


Fig. 22. Displacement field from FEM (Poisson's ratio = 0.45).

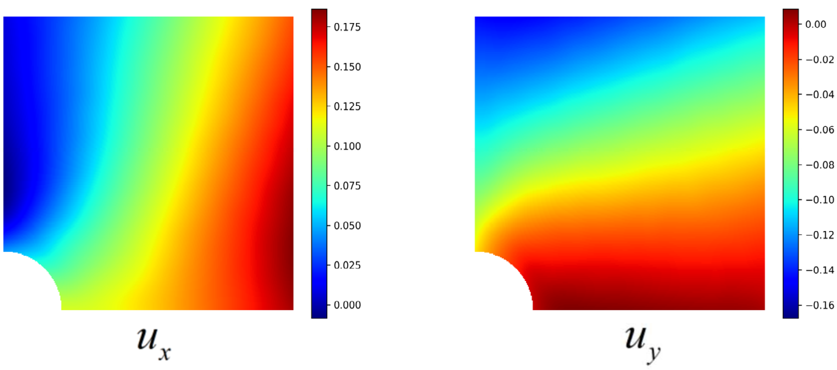
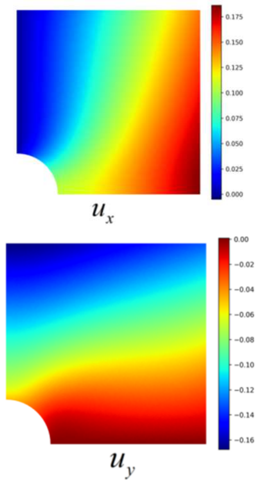
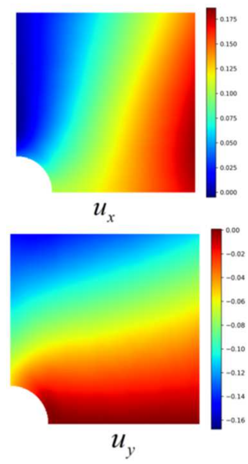
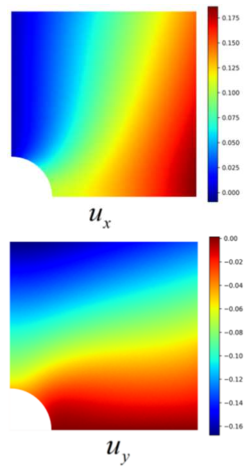
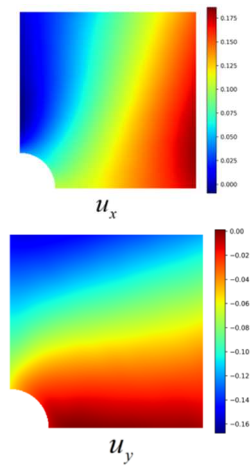
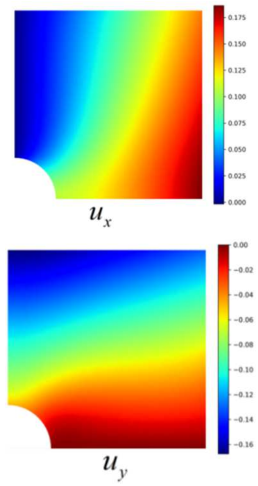
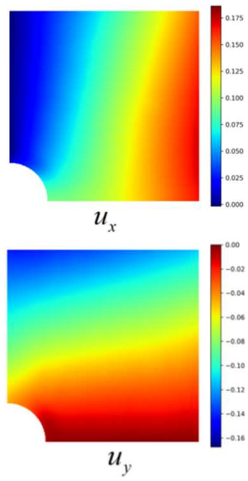
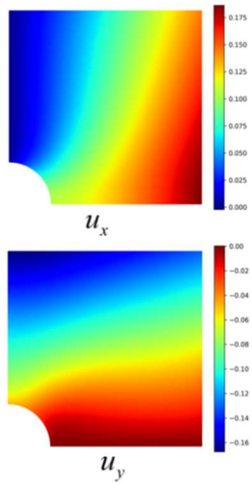
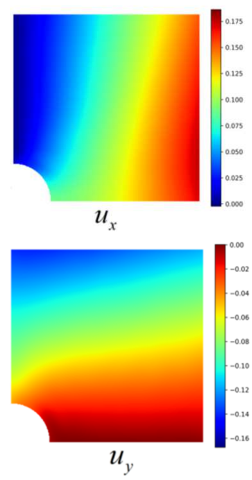
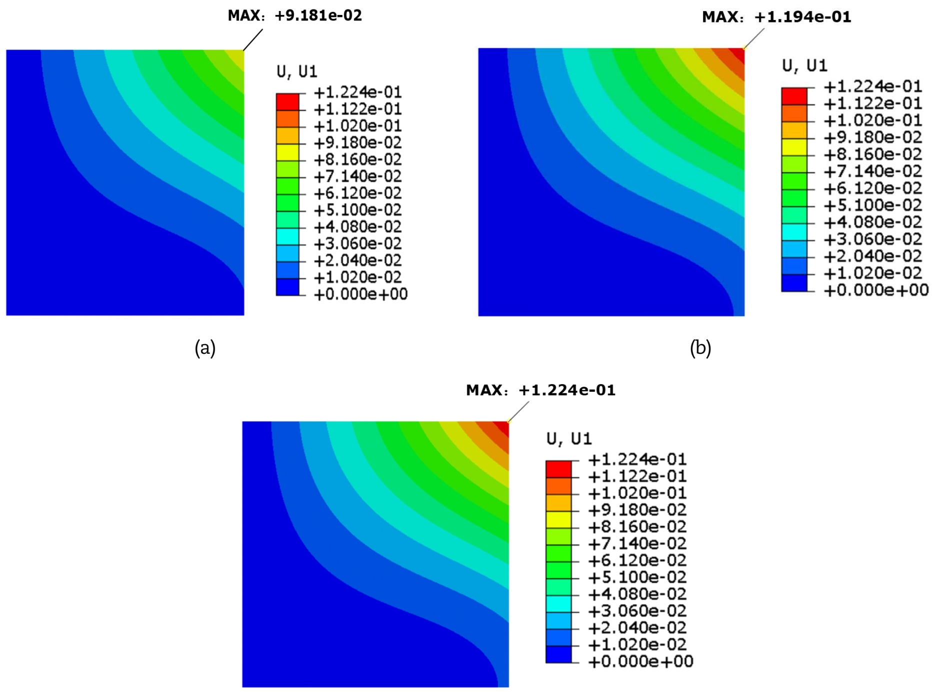


Fig. 23. Displacement field from PINN (Poisson's ratio = 0.45).

Table 3. Comparison of results under different Poisson's ratios.

Poisson's Ratio μ	FEM Results	PINN Results	Poisson's Ratio μ	FEM Results	PINN Results
μ = 0.46			μ = 0.47		
μ = 0.48			μ = 0.49		



(a) (b) (c)

Fig. 24. Comparison of results using different finite element types, (a) Full integration element FEM results, (b) Reduced integration element FEM results, (c) Global reduced integration element with hourglass control FEM results.

Table 4. Comparison of horizontal displacement u_x across finite element types.

	Element Type	Horizontal Displacement u_x	Analytical Solution	Error
FEM model 1	Full Integration Element	0.0918	0.125	26.56%
FEM model 2	Reduced Integration Element	0.1194	0.125	4.48%
FEM model 3	Reduced Integration with Hourglass Control	0.1224	0.125	2.08%
Mean Squared Error (MSE) -based PINN model	/	0.1238	0.125	0.97%
Energy-based PINN model	/	0.1236	0.125	1.12%

4.4. Discussion on finite element types

To investigate how finite element type selection affects the mitigation of volumetric locking in FEM, this section compares different element types using the plane strain problem from Example 3. The Abaqus was used to simulate the structure with: (1) full integration elements, (2) reduced integration elements, and (3) reduced integration elements with hourglass control, all at Poisson's ratio of 0.45. The results are shown in Fig. 24, with PINN method results referenced from Fig. 18.

For clearer comparison of results across element types, this section uses the horizontal displacement u_x at point (1,1) (location of maximum displacement) to evaluate calculation accuracy through nodal displacement. Meanwhile, the results obtained from both the energy-based and Mean Squared Error (MSE) -based PINN models are compared with the theoretical solution, as summarized in Table 4.

As shown in Table 4, FEM model 1 using full integration elements exhibits significant errors compared to other models, with completely distorted displacement accuracy at nodes, clearly demonstrating volumetric locking phenomena. This leads to the conclusion that finite element methods using full integration elements cannot properly simulate the mechanical behavior of nearly incompressible materials. FEM model 2 using reduced integration elements shows results closer to the theoretical solution, effectively mitigating volumetric locking. FEM model 3, employing reduced integration elements with hourglass control, produces displacement solutions even closer to the theoretical values, demonstrating good solution accuracy and successful elimination of volumetric locking. In comparison with the three finite element models, the displacement solutions obtained from both PINN approaches are highly consistent with the theoretical solution. Notably, the PINN method achieves this accuracy without requiring mesh generation or complex finite elements, thereby significantly improving computational efficiency. These results confirm that the ability to avoid locking is an inherent feature of the mesh-free continuous solution paradigm of PINNs.

5. Influence of Neural Network Hyperparameters on Convergence

In Physics-Informed Neural Networks (PINN), the hyperparameters of the neural network not only determine the approximation capability of the differential equation solutions but also play a key role in embedding physical constraints. Unlike traditional neural networks, PINN optimization needs to simultaneously minimize both data-fitting errors and physical law violations, making the careful selection of neural network parameters essential. The following section uses Example 1 from Section 3.3.1 to explore how to choose the appropriate neural network parameters. This includes the number of hidden layers and neurons, activation function, learning rate, sampling method, loss function, and optimizer. The aim is to identify the most suitable neural network parameters for the given model.

5.1. Number of hidden layers and neurons

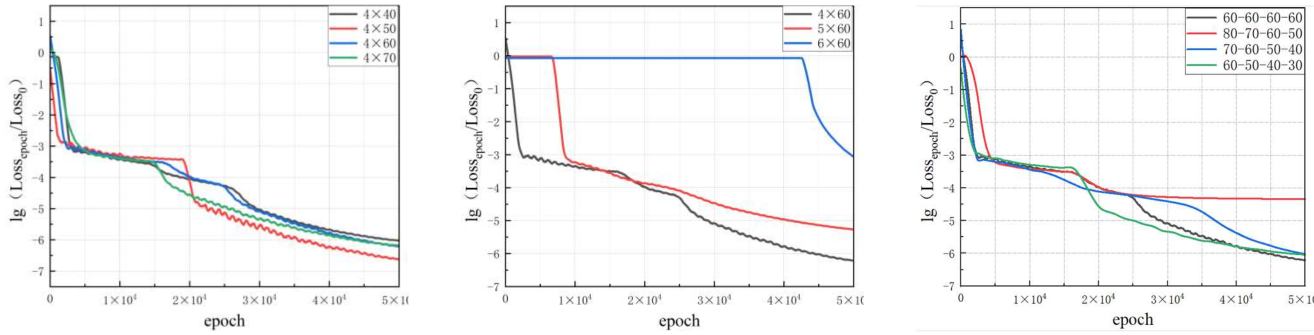
In the actual training process, the number of hidden layers and neurons significantly affects the training efficiency and convergence of the neural network. Too many hidden layers can lead to issues such as vanishing or exploding gradients, making the training process difficult or even causing it to fail to converge. On the other hand, too few hidden layers may limit the model's ability to learn from the data, leading to underfitting. Likewise, too many neurons can result in overfitting, where the model performs well on the training set but has poor generalization ability on the test set. Conversely, too few neurons restrict the model's fitting capability, leading to underfitting. Therefore, when designing a neural network model, it is essential to choose an appropriate hidden layer structure based on the complexity of the problem to achieve effective convergence of the loss function and obtain good model performance.

As shown in Fig. 25, the x-axis represents the training epochs, while the y-axis represents the logarithm of the ratio of the current epoch's loss function ($Loss_{epoch}$) to the initial loss function value ($Loss_0$) before training. For the chosen example, 4 hidden layers with approximately 60 neurons per layer achieve optimal precision. A model with 50 neurons exhibits large fluctuations, while a model with 60 neurons shows smaller fluctuations and a faster rate of loss reduction compared to one with 70 neurons. Therefore, 60 neurons were selected. Overall, the selection of the number of neurons and hidden layers is highly empirical, requiring repeated adjustments to balance both parameters to achieve the best model accuracy. Additionally, as shown in Fig. 25(b), too large a number of hidden layers can slow down the rate of loss reduction, and it does not necessarily improve computational accuracy.

Furthermore, a comparison of different hidden layer configurations with the same number of neurons is presented in Fig. 25(c). It can be observed that in the case of the plane stress problem (Example 1), selecting the same number of neurons across different hidden layers yields better results. Moreover, using the same number of neurons simplifies the network and ensures symmetry, allowing each layer to extract the same number of features from the previous layer, which also benefits the stability of the network.

5.2. Activation function

The choice of activation function significantly impacts the convergence of neural networks. Different activation functions affect the performance and convergence speed of neural networks. Choosing the appropriate activation function can improve training efficiency and generalization capability of the model [35]. The PINN model in this paper incorporates four commonly used activation functions: ReLU, Sigmoid, Tanh, and Softplus. By defining the neural network model, an appropriate activation function can be selected based on the problem type to enhance performance and convergence speed. The ReLU function is typically used in hidden layers, especially in deep neural networks, where it effectively mitigates the vanishing gradient problem. The Sigmoid function is used in the output layer for binary classification problems or in shallow neural networks. The Tanh function converges faster and is typically used in hidden layers, offering stronger non-linearity. The Softplus function is commonly used in hidden layers, providing smooth non-linearity. In the analysis of the selected example, the Tanh function was chosen as the activation function for the model.



(a) (b) (c)

Fig. 25. Influence of different network parameters on the convergence characteristics of loss function, (a) Comparison of loss of different neuron numbers, Comparison of loss in different numbers of hidden layers, (c) Comparison of the number of neurons in different hidden layers.

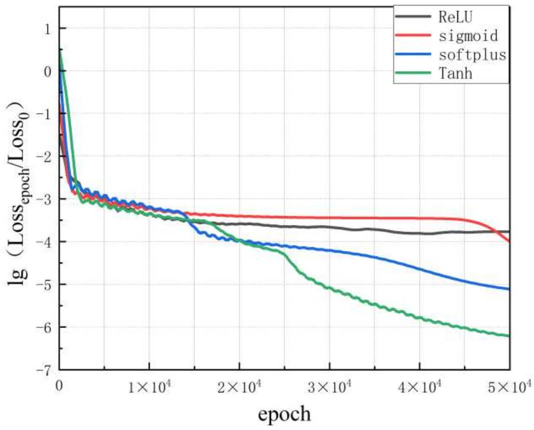


Fig. 26. Influence of different activation functions on the convergence characteristics of loss function.

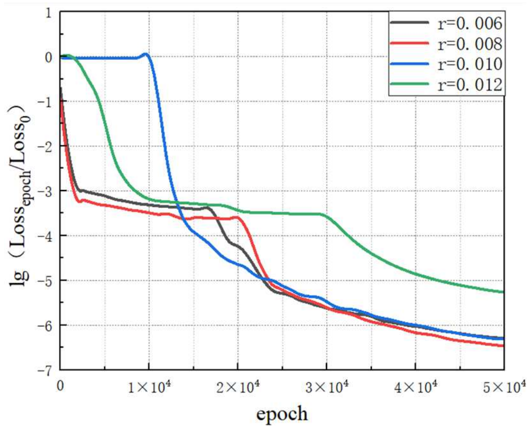


Fig. 27. Influence of different learning rates on the convergence characteristics of the loss function.

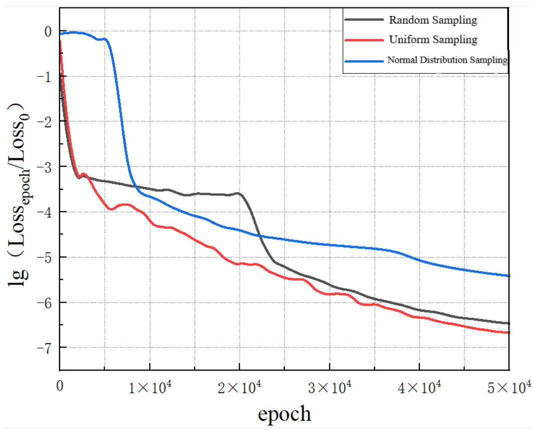


Fig. 28. Influence of different sampling methods on the convergence characteristics of the loss function.

As shown in Fig. 26, using the Tanh activation function achieves the best accuracy for the above example. The results obtained with the Tanh activation function are clearly superior to the other three activation functions, showing excellent performance in terms of loss reduction speed.

5.3. Learning rate

The learning rate is a critical hyperparameter in neural network training, and its selection directly affects both the convergence speed and the stability of the training process. If the learning rate is too large, the loss will oscillate dramatically and fail to converge, potentially increasing the loss further. If the learning rate is too small, the model parameters will update too slowly, making the training process slower and requiring more iterations to achieve the same performance [36]. Therefore, when designing a neural network model, it is essential to select an appropriate learning rate based on the specific problem to ensure that the loss decreases smoothly during training.

The examples in this study are all static problems. To improve computational efficiency, a constant learning rate, commonly used in neural networks, is employed in the network structure. As shown in Fig. 27, when the learning rates are set to 0.006, 0.008, and 0.010, the convergence behavior of the loss function is favorable. As the number of iterations increases, the convergence speed for a learning rate of 0.006 is slower than that of 0.008, while the results for a learning rate of 0.010 show lower accuracy than those for 0.008. Therefore, a learning rate of 0.008 is the optimal choice for this example.

5.4. Sampling method

The method of selecting interior points in the neural network significantly impacts the solution accuracy of the PINN model. Therefore, an appropriate sampling method must be chosen based on the specific problem to ensure optimal solutions.

As shown in Fig. 28, three sampling methods—random sampling, normal distribution sampling, and uniform sampling—were used to train and compare the model for the selected example. The normal distribution sampling requires selecting points in the x-direction with a mean of 1/2 and a standard deviation of 1/6, and in the y-direction with a mean of 0.1/2 and a standard deviation of 0.1/6. Uniform sampling involves evenly spaced point selection in the x-direction [0, 1] and y-direction [-0.05, 0.05].

It can be observed that both random sampling and uniform sampling methods yield excellent results. To enhance the model's generalization ability and enable it to adapt to complex structures, random sampling was chosen as the sampling method for the selected example.

5.5. Loss function

In the training process of a neural network model, the loss function plays a central role in the entire network architecture. The choice of loss function directly affects the stability of the optimization process and the accuracy of the final solution. This section analyzes and compares three commonly used loss functions in the neural network field: Mean Squared Error Loss (MSELoss), Mean Absolute Error Loss (MAELoss), and Smooth L1 Loss (SmoothLoss). By employing different loss functions in the neural network, the results are compared to investigate their impact on the network's performance.

The Mean Squared Error Loss (MSELoss) measures the error by calculating the sum of the squared differences between the predicted values and the true values. It assigns a higher penalty weight to larger errors, thus focusing the training process on reducing significant deviations. For the pure bending beam stress model, the corresponding loss function is given by equation (31):

$$\begin{cases} MSE = \varphi_{PDE} MSE_{PDE} + \varphi_{BC} MSE_{BC} \\ MSE_{PDE} = MSE_{P} + MSE_{G} + MSE_{C} \end{cases} \tag{31}$$

The Mean Absolute Error Loss (MAELoss) measures the error by calculating the sum of the absolute differences between the predicted values and the true values. It assigns linear weights to all error terms, which makes it more robust to outliers. For the pure bending beam stress model, the corresponding loss function is given by Eq. (32), where MAEPDE is the mean absolute error of the partial differential equation, MAEBC is the mean absolute error of the boundary conditions, and MAEP, MAEG, MAEC represent the mean absolute errors of the governing equations, geometric equations, and constitutive relations, respectively:

$$\begin{cases} MAE = \varphi_{PDE} MAE_{PDE} + \varphi_{BC} MAE_{BC} \\ MAE_{PDE} = MAE_{P} + MAE_{G} + MAE_{C} \end{cases} \tag{32}$$

The Smooth L1 Loss (SmoothLoss) can be seen as a compromise between MSELoss and L1 Loss. It uses a squared term for small errors to ensure differentiability and switches to a linear term for larger errors to improve robustness. For the pure bending beam stress model, the corresponding loss function is given by equation (33), where $Smooth_{PDE}$ is the smooth L1 loss for the partial differential equation, $Smooth_{BC}$ is the smooth L1 loss for the boundary conditions, and $Smooth_{P}$, $Smooth_{G}$, $Smooth_{C}$ represent the smooth L1 losses for the governing equations, geometric equations, and constitutive relations, respectively:

$$\begin{cases} Smooth = \varphi_{PDE} Smooth_{PDE} + \varphi_{BC} Smooth_{BC} \\ Smooth_{PDE} = Smooth_{P} + Smooth_{G} + Smooth_{C} \end{cases} \tag{33}$$

As shown in Fig. 29, MAELoss and SmoothLoss have slower convergence rates compared to MSELoss. Due to the squared nature of MSELoss, it has the advantage of continuous differentiability, providing stable and consistent gradient signals in gradient descent optimization. This makes MSELoss suitable for scenarios that require precise fitting of smooth functions, giving MSELoss a significant advantage in the pure bending beam stress model.

5.6. Optimizer

In the training process of the PINN, the choice of optimization algorithm plays a decisive role in the model's convergence characteristics and final solution accuracy. This study systematically compares the performance of five optimizers—Adagrad, Adadelta, Adam, Adamax, and L-BFGS—in solving plane problems. It reveals the unique behaviors of each optimization algorithm in the context of physics-informed embedding, and selects the most suitable optimizer for the research case based on the comparative analysis.

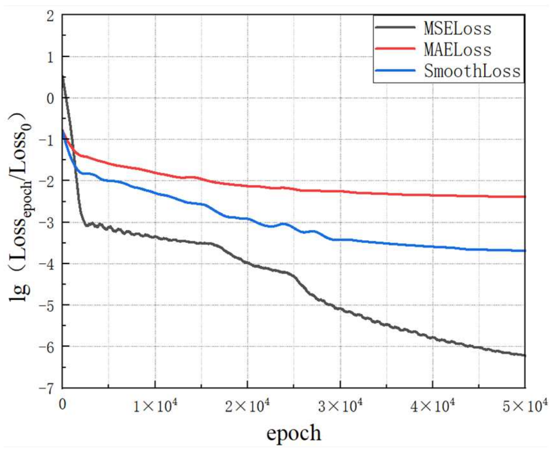


Fig. 29. Influence of different loss functions on loss convergence characteristics.

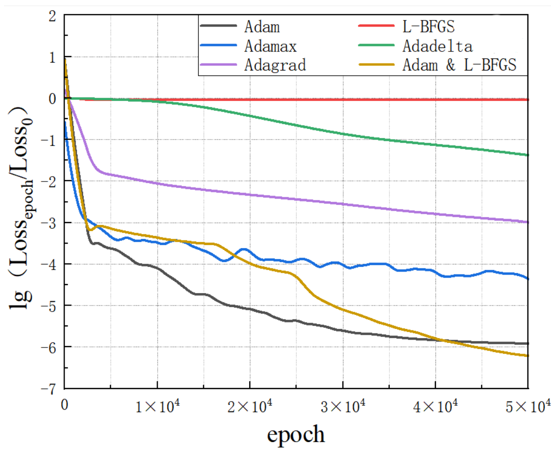


Fig. 30. Influence of different optimizers on loss convergence characteristics.

As shown in Fig. 30, the Adam optimizer outperforms the other four optimizers, demonstrating superior performance. The Adam optimizer accelerates initial convergence through its unique momentum mechanism, providing personalized update steps for each parameter. This results in faster convergence compared to other optimizers, enabling a quicker reduction in loss to a relatively optimal range. In contrast, the L-BFGS optimizer behaves quite differently, with its optimization speed significantly slower than the other optimizers. This is because L-BFGS is highly sensitive to the initial learning rate, with its convergence speed heavily dependent on the quality of the initial parameter values. Moreover, it uses explicit gradient calculations multiple times to evaluate the optimization process. Consequently, the L-BFGS optimizer is more suited for high-precision problem-solving.

To further discuss the selection of optimizers, this section combines the strengths of the Adam and L-BFGS optimizers and compares them with the other optimizers. By utilizing the complementary advantages of both optimizers in different stages, this approach achieves a synergistic improvement in both training efficiency and solution accuracy. With the hybrid optimizer, the Adam optimizer is used initially to quickly locate the optimal solution neighborhood, and later, the L-BFGS optimizer is employed to fine-tune the solution using its quasi-Newton method for second-order convergence. This results in a superior final outcome compared to using only the Adam optimizer, with a convergence speed much higher than when using L-BFGS alone. As seen in Fig. 30, the results from the hybrid optimizer are consistently better than those from the other optimizers.

This section analyzes the impact of optimizer selection on the performance of various optimizers and, through the combination of two optimizers, identifies the optimal optimizer for the neural network model in this study.

6. Conclusions

This study, based on the Physics-Informed Neural Network (PINN) method, solves two types of plane problems while avoiding the occurrence of locking phenomena in Finite Element Methods (FEM). The method's effectiveness has been validated through typical test cases. Based on the comprehensive research results, the following conclusions can be drawn:

1) The PINN method proposed in this study effectively solves plane problems without requiring any labeled data, relying solely on governing equations and boundary conditions. More importantly, by constructing and comparing models based on both the strong-form PDE and the weak-form energy principle in parallel, this work elucidates that the intrinsic reason PINNs can avoid locking lies in their paradigm of *continuous function approximation* and *mesh-free global optimization*, rather than in any particular implementation. This finding provides theoretical depth and stronger evidence beyond numerical examples for the assertion that “PINNs inherently possess a locking-free property.”
2) Compared with the conventional finite element method, the proposed approach eliminates the influence of mesh partitioning on computational efficiency and accuracy. By leveraging mesh-free modeling and the embedding of physical information—whether through the strong-form PDE or the weak-form energy principle—it fundamentally avoids numerical deficiencies caused by element discretization and low-order interpolation. For plane stress problems, it circumvents shear locking without considering mesh effects, achieving high-accuracy solutions solely by optimizing the loss function. For plane strain problems, it avoids volumetric locking induced by mesh discretization in FEM, without relying on complex techniques such as mixed elements or reduced integration, thereby demonstrating a unique advantage in handling strongly constrained conditions. In both cases, the mesh-free nature significantly simplifies the computational process while ensuring accuracy.
3) In the plane stress problem case, the results exhibit minimal error compared to the FEM solution with 100,000 mesh elements, and the strong form governing equations strictly adhere to the physical laws, successfully avoiding shear locking. In the plane strain problem case, for incompressible materials with a Poisson's ratio close to 0.5, the strain field distribution closely matches the theoretical solution, with no volumetric locking phenomenon. This demonstrates that the PINN method, by embedding physical information, can simultaneously address both types of locking problems and achieve higher accuracy than traditional numerical methods. This feature highlights the method’s effectiveness and precision.
4) The contribution of this work lies not only in the successful application of PINNs to solve two classes of plane problems, but also in demonstrating, through a comparative study of two forms of physical constraints, the intrinsic necessity and universality of mesh-free neural network methods in avoiding numerical locking. This study deepens the understanding of the advantages of PINNs in computational mechanics, offers a new perspective on how neural-network-based numerical methods overcome the theoretical limitations of traditional FEM, and lays a solid foundation for addressing broader locking issues, such as shear locking in plate and shell analysis.

Through its mesh-free nature, embedded physical information, and adaptive optimization characteristics, the PINN method effectively avoids shear locking in plane stress problems and volumetric locking in plane strain problems, while maintaining high precision and computational efficiency. This study provides a new paradigm for the numerical solution of solid mechanics problems, offering significant theoretical and engineering value.

Author Contributions

Z. Lin designed the study, developed the neural network model, conducted the experiments, and analyzed the results; K. Du proposed experimental suggestions, initiated the project, and contributed to the modification and optimization of the neural network model; Z Li carried out the theoretical validation and verified the experimental results. This manuscript was completed through the joint efforts of all authors. All authors discussed the results, reviewed the manuscript, and approved its final version.

Acknowledgments

The authors would like to express their sincere gratitude to the two anonymous reviewers for their thorough evaluation of this work and for providing valuable insights and constructive comments. The authors also extend their thanks to Editor-in-Chief Hamid M. Sedighi for his support in preparing the final version of this article for publication.

Conflict of Interest

The authors declared no potential conflicts of interest concerning the research, authorship, and publication of this article.

Funding

This study was supported by funds from the Scientific Research Fund of the Institute of Engineering Mechanics, China Earthquake Administration (Grant No. 2026C02) and the National Key Research and Development Program of China (Grant No. 2024YFC3012802).

Data Availability Statements

The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

[1] Xue, J., Deng, Z., Finite element analysis on elastic plate bending, *Journal of Shenyang Aerospace University*, 28(4), 2011, 9-11.
[2] Dang, X., Lu, X., Qian, J., Jiang, H., Finite element analysis with solid and plane element of seismic performance of self-centering pre-stressed shear walls, *Journal of Building Structures*, 35(5), 2014, 17-24.
[3] Cao, Z., Guan, Y., Chen, W., A novel 2-dimensional hybrid stress element for stress analyses of plate with holes, *Chinese Journal of Solid Mechanics*, 36(4), 2015, 329-336.
[4] Zheng, Y., Zhao, S., Cheng, C., Development of finite element limit analysis method and its applications in geotechnical engineering, *Strategic Study of CAE*, 12, 2006, 39-61+112.
[5] Wang, L., Zhang, X., Tinti, S., Large deformation dynamic analysis of progressive failure in layered clayey slopes under seismic loading using the particle finite element method, *Acta Geotechnica*, 16(8), 2021, 2435-2448.
[6] Monforte, L., Arroyo, M., Carbonell, J.M., Gens, A., Coupled effective stress analysis of insertion problems in geotechnics with the particle finite element method, *Computers and Geotechnics*, 101, 2018, 114-129.
[7] Monforte, L., Carbonell, J.M., Arroyo, M., Gens, A., Performance of mixed formulations for the particle finite element method in soil mechanics problems, *Computational Particle Mechanics*, 4(3), 2017, 269-284.
[8] Monforte, L., Arroyo, M., Carbonell, J.M., Gens, A., Numerical simulation of undrained insertion problems in geotechnical engineering with the particle finite element method (PFEM), *Computers and Geotechnics*, 82, 2017, 144-156.
[9] Zhang, X., Krabbenhoft, K., Sheng, D.C., Li, W.C., Numerical simulation of a flow-like landslide using the particle finite element method, *Computational Mechanics*, 55(1), 2015, 167-177.
[10] Zhang, X., Krabbenhoft, K., Sheng, D.C., Particle finite element analysis of the granular column collapse problem, *Granular Matter*, 16(4), 2014, 609-619.
[11] Wang, Z., Kong, D., Du, Y., Zhu, B., Extension and validation of sequential limit analysis for geotechnical problems, *Rock and Soil Mechanics*, 44(12), 2023, 3531-3540.
[12] Nayroles, B., Touzot, G., Villon, P., Generalizing the finite element method: Diffuse approximation and diffuse elements, *Computational Mechanics*, 10(5), 1992, 307-318.
[13] Belytschko, T., Gu, L., Lu, Y.Y., Fracture and crack growth by element free Galerkin methods, *Modelling and Simulation in Materials Science and Engineering*, 2(3A), 1994, 519-534.
[14] Cho, J.Y., Atluri, S.N., Analysis of shear flexible beams, using the meshless local Petrov-Galerkin method, based on a locking-free formulation, *Engineering Computations*, 18(1/2), 2001, 215-240.
[15] Noguchi, H., Kawashima, T., Miyamura, T., Element free analyses of shell and spatial structures, *International Journal for Numerical Methods in Engineering*, 47(6), 2015, 1215-1240.
[16] Kanok-Nukulchai, W., Barry, W., Saran-Yasoontorn, K., Bouillard, P.H., On elimination of shear locking in the element-free Galerkin method, *International Journal for Numerical Methods in Engineering*, 52(7), 2001, 705-725.
[17] Donning, B.M., Liu, W.K., Meshless methods for shear deformable beams and plates, *Computer Methods in Applied Mechanics and Engineering*, 152, 1998, 47-71.
[18] Chen, J., Wu, C., Yoon, S., A stabilized conforming nodal integration for Galerkin mesh-free methods, *International Journal for Numerical Methods in Engineering*, 50(2), 2001, 435-466.
[19] Wang, D., Chen, J.S., Locking-free stabilized conforming nodal integration for meshfree Mindlin–Reissner plate formulation, *Computer Methods in Applied Mechanics and Engineering*, 193(12–14), 2004, 1065-1083.
[20] Krizhevsky, A., Sutskever, I., Hinton, G., ImageNet classification with deep convolutional neural networks, *Communications of the ACM*, 60(6), 2017, 84-90.
[21] Kissas, G., Yang, Y., Hwuang, E., et al., Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks, *Computer Methods in Applied Mechanics and Engineering*, 2019, 112623.
[22] Zhang, R., Liu, Y., Sun, H., Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling, *Engineering Structures*, 215, 2020, 110704.
[23] Su, G., Song, Y., Yan, L., Application of Gaussian process machine learning to slope stability evaluation, *Rock and Soil Mechanics*, 30(3), 2009, 675-679.
[24] Chen, C., *Detection method of thermal engineering defects in exterior thermal insulation buildings based on machine learning*, Ph.D. Thesis, Harbin Institute of Technology, Harbin, China, 2019.
[25] Raissi, M., Perdikaris, P., Karniadakis, G.E., Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics*, 378, 2019, 686-707.
[26] Wang, J.X., Wu, J.L., Xiao, H., Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data, *Physical Review Fluids*, 2(3), 2017, 1-22.
[27] Wang, B., Ren, H.J., Cao, W.X., Hu, Y.W., He, Y.R., Han, C.J., Zhu, J.Q., Prediction of thermal conductivity of diamond film by neural network based on first principles, Science China Physics, *Mechanics & Astronomy*, 52(9), 2022, 62-74.
[28] Li, W., Bazant, M.Z., Zhu, J., A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches, *Computer Methods in Applied Mechanics and Engineering*, 383, 2021, 113933.
[29] Bai, J., Rabczuk, T., Gupta, A., et al., A physics-informed neural network technique based on a modified loss function for computational 2D and 3D solid mechanics, *Computational Mechanics*, 71, 2023, 543–562.
[30] Wang, Y., Sun, J., Rabczuk, T., et al., DCEM: A deep complementary energy method for linear elasticity, *International Journal for Numerical Methods in Engineering*, 125(24), 2024, e7585.
[31] Huang, H.C., Implementation of assumed strain degenerated shell elements, *Computers and Structures*, 25(1), 1987, 147-155.
[32] Yu, Y., Yao, H., Liu, Y., Structural dynamics simulation using a novel physics-guided machine learning method, *Engineering Applications of Artificial Intelligence*, 96, 2020, 103947.
[33] Li H., Miao Y., Khodaei Z.S., Aliabadi, M.H., Finite-PINN: A physics-informed neural network with finite geometric encoding for solid mechanics, *Journal of the Mechanics and Physics of Solids*, 203, 2025, 106222.
[34] Samaniego, E., Anitescu, C., Goswami, S., et al., 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, 2020, 112790.
[35] Zhang, Y.J., Chen, C., Wang, J.J., Research on activation functions of deep learning algorithms, *Radio Communication Technology*, 47(1), 2021, 115-120.
[36] Li, Y.R., Zhang, Y.L., Wang, J.C., Survey on Bayesian optimization methods for hyper-parameter tuning, *Computer Science*, 49(S1), 2022, 86-92.

ORCID iD

Zhipeng Lin https://orcid.org/0009-0003-5375-9504
Ke Du https://orcid.org/0000-0002-4793-0542
Zepeng Li https://orcid.org/0009-0005-2381-4538

How to cite this article: Lin Z., Du K., Li Z. A Mesh-Free Physics-Informed Neural Network Framework for Solving Plane Problems without Locking Issues, *J. Appl. Comput. Mech.*, 12(3), 2026, 1047-1068. https://doi.org/10.22055/jacm.2025.48578.5339