Energy-Based Physics-Informed Neural Network for Linear Elastic Analysis of Multi-patch Plate and Shell Structures

GUO Boyu[1], *CHEN Yuhang*[2], *QIU Hao*[1], *XU Xianhua*[1], *GUO Yujie*[1*]

1. College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P. R. China;
2. China Aerodynamics Research and Development Center, Mianyang 621000, P. R. China

(Received 19 December 2025; revised 16 March 2026; accepted 20 April 2026)

Abstract: Machine learning provides a fast and accurate tool for the prediction of a physical model. In this paper, a machine learning framework based on the physics-informed neural network (PINN) was established to predict the linear elastic static deformation of plate and shell structures. In contrast to the purely data-driven neural network, PINN incorporates the physical laws into the training process, thus reducing the required amount of data. The loss functions of the PINN are constructed based on the total potential energy functions of the thin-walled structure. Besides, the proposed PINN can be easily extended to shell structures with multiple patches by adding interface compatibility constraints into the loss function. The performance of the PINNs with the energy-based loss functions was evaluated with different shell structures and compared with the finite element results. Numerical examples show that the highly accurate results can be achieved based on the proposed framework which significantly reduces the amount of required training data compared to the data-driven neural network.

Key words: physics-informed neural network (PINN); machine learning; plate and shell structures; total potential energy; partial differential equations

CLC number: O342 **Document code:** A **Article ID:** 1005-1120(2026)03-0463-12

0 Introduction

Shell structures are widely used in aerospace, ship building and automobile industries. The prediction of mechanical responses of thin shell structures has received considerable attention. Traditional approaches such as finite element methods (FEMs) are widely used for the analysis of shell structures. But, for more complex problems including nonlinearities and dynamic behaviors, FEM is still time consuming. Therefore, numerical methods with fast evaluation properties, such as reduced order model, machine learning, etc., have received increasing attention over the past decade[1-2].

The construction of a reliable machine learning scheme relies on vast amount of data, the appropriate model and proper training approach. Typical machine learning method such as artificial neural network (ANN) is based on the human brain perceptron mechanism. It is demonstrated that, with sufficient data and appropriate training, ANN can predict rather accurate results. However, when the training data are not adequate, it may deteriorate the effectiveness of the ANN[3].

Physics-informed neural network (PINN) was initially proposed in Ref.[4] where initial/boundary conditions were included in the model to solve ordinary and partial differential equations. It then gained particular attention due to the advances of computing power[5]. PINNs incorporate the already-known physical laws into the process of training, thus reduceing the amount of training data significantly. Raissi et al.[5] introduced the partial differential equations (PDEs) and the associated initial and boundary con-

*Corresponding author, E-mail address: yujieguo@nuaa.edu.cn.

How to cite this article: GUO Boyu, CHEN Yuhang, QIU Hao, et al. Energy-based physics-informed neural network for linear elastic analysis of multi-patch plate and shell structures[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2026, 43(3):463-474.

http://dx.doi.org/10.16356/j.1005-1120.2026.03.010

ditions into the loss function and solved the fluids, quantum mechanics and reaction-diffusion problems. In Ref.[6], a deep learning Ritz method was proposed, where the loss function is defined by the energy functional corresponding to the PDEs. It is found that, the proposed method can also identify the unknown parameters in a physical law.

For thin shell problems, the physical quantities are tensorial components, and the number of variables is much higher than the other problems, which brings extra difficulties in constructing ANN. Besides, the governing equations of thin shells have high order derivatives, which needs extra computational efforts and may decrease the computational efficiency. Recently, Li et al.[7] developed a PINN for elastic plate structures, and systematically compared the effectiveness of the PDE-based and energy-based loss functions. In addition, Bastek et al.[8] extended the PINN framework to the analysis of shell structures and studied the influence of the small-thickness limit on the accuracy of the PINN. Besides, Yan et al.[9] extended the concept of PINN to the composite plates and the shallow cylindrical shells, where extreme learning machines was proposed to accelerate the training process. Other work related to the structural analysis using PINN can be found in Refs.[10-12], to name a few.

PINNs have also been actively developed for a broad range of scientific and engineering problems beyond structural mechanics, including transport-type problems and other coupled PDE systems. Recent studies have reported methodological advances such as enriched/modified PINN formulations and improved training strategies (e.g., adaptive loss balancing), as well as hybrid PINN frameworks combined with classical numerical techniques for boundary-value and inverse problems. These developments further demonstrate the rapid and dynamic progress of PINNs and provide a broader context for the present work[13-15].

Our multi-patch strategy is conceptually related to domain-decomposition PINN frameworks such as conservative PINNs (cPINNs) and extended PINNs (XPINNs), which decompose the computational domain into multiple subdomains and use separate neural networks together with interface conditions to couple the subproblems. However, the present work is formulated in an energy-based manner, where the loss function is constructed from the total potential energy functional and the coupling between patches is imposed through compatibility constraints. In contrast to strong-form XPINN/cPINN approaches based on PDE residuals, the proposed weak-form energy-based formulation avoids the explicit evaluation of high-order derivatives, which is particularly advantageous for thin-walled plate and shell problems and improves the robustness of the multi-patch analysis[16-17].

In this paper, we develop an energy-based PINN framework for the linear elastic static analysis of thin-walled plate and shell structures. The proposed PINN uses the total potential energy functional as the loss function, thus avoiding the explicit evaluation of high-order derivatives required in strong-form PINNs. Furthermore, essential boundary conditions and patch compatibility constraints can be naturally incorporated into the loss function, facilitating the modeling and analysis of complex shell structures.

The main contributions of this work can be summarized as follows:

(1) An energy-based PINN framework is proposed for the linear elastic static analysis of thin-walled plate and shell structures, improving numerical robustness by avoiding high-order derivatives.

(2) The proposed method is validated on representative plate and shell benchmarks, showing excellent agreement with FEM reference solutions.

(3) The framework is extended to multi-patch shell structures by introducing interface compatibility constraints, enabling the efficient analysis of complex geometries such as wing box structures.

(4) The results demonstrate that high prediction accuracy can be achieved with significantly reduced training data compared to purely data-driven neural networks.

The paper is organized as follows. The governing equations of thin-shell structures and the constructions of the physics-informed neural network are presented in Section 1. The numerical examples are given in Section 2. Finally, we summarize the main findings and draw conclusions in Section 3.

1 Theories

In this section, we first introduce the basic equations governing the behavior of thin shells in Section 1.1. Then, we briefly introduce the underlying theories of the physics-informed neural network in Section 1.2.

1.1 Governing equations of thin shells

The total potential energy of the thin shell is expressed as

$$\Pi = U + V \tag{1}$$

where U is the strain energy stored in the shell, it can be expressed as $U = U_{\mathrm{m}} + U_{\mathrm{b}}$, where U_{m} and U_{b} are the membrane and bending strain energy which can be expressed, respectively, shown as

$$U_{\mathrm{m}} = \iint_S \frac{C}{2}\left((\varepsilon_\alpha + \varepsilon_\beta)^2 + 2(1-\mu)\left(\frac{1}{4}\gamma_{\alpha\beta}^2 - \varepsilon_\alpha\varepsilon_\beta\right)\right)AB\mathrm{d}\alpha\mathrm{d}\beta \tag{2}$$

$$U_{\mathrm{b}} = \iint_S \frac{D}{2}\left((\kappa_\alpha + \kappa_\beta)^2 + 2(1-\mu)\left(\frac{1}{4}\chi_{\alpha\beta}^2 - \kappa_\alpha\kappa_\beta\right)\right)AB\mathrm{d}\alpha\mathrm{d}\beta \tag{3}$$

and V in Eq.(1) is the work done by external forces, which can be written as

$$V = -\int_S \boldsymbol{P}\cdot\boldsymbol{u}\mathrm{d}S - \int_\Gamma \boldsymbol{T}\cdot\boldsymbol{u}\mathrm{d}\Gamma \tag{4}$$

where $\boldsymbol{P}$ is the distributed force vector on the shell surface S, $\boldsymbol{T}$ the traction at the boundary Γ, and $\boldsymbol{u}$ the displacement vector of the shell

The quantities C and D in Eqs.(2, 3) are the stretching and bending stiffness, shown as

$$C = \frac{Eh}{1-\mu^2}, \quad D = \frac{Eh^3}{12(1-\mu^2)} \tag{5}$$

where E is the Young's modulus, h the thickness of the shell, and μ the Poisson's ratio.

The membrane strain components in Eq.(2) can be represented as

$$\varepsilon_\alpha = \frac{1}{A}\frac{\partial u_\alpha}{\partial\alpha} + \frac{u_\beta}{AB}\frac{\partial A}{\partial\beta} + \frac{u_n}{R_1} \tag{6}$$

$$\varepsilon_\beta = \frac{1}{B}\frac{\partial u_\beta}{\partial\beta} + \frac{u_\alpha}{AB}\frac{\partial B}{\partial\alpha} + \frac{u_n}{R_2} \tag{7}$$

$$\gamma_{\alpha\beta} = \frac{B}{A}\frac{\partial}{\partial\alpha}\left(\frac{u_\beta}{B}\right) + \frac{A}{B}\frac{\partial}{\partial\beta}\left(\frac{u_\alpha}{A}\right) \tag{8}$$

The bending strain components in Eq.(3) can be represented as[18]

$$\kappa_\alpha = \frac{1}{A}\frac{\partial}{\partial\alpha}\left(\frac{u_\alpha}{R_1}\right) + \frac{1}{AB}\frac{\partial A}{\partial\beta}\frac{u_\beta}{R_2} - \frac{1}{A}\frac{\partial}{\partial\alpha}\left(\frac{1}{A}\frac{\partial u_n}{\partial\alpha}\right) - \frac{1}{AB^2}\frac{\partial A}{\partial\beta}\frac{\partial u_n}{\partial\beta} \tag{9}$$

$$\kappa_\beta = \frac{1}{B}\frac{\partial}{\partial\beta}\left(\frac{u_\beta}{R_2}\right) + \frac{1}{AB}\frac{\partial B}{\partial\alpha}\frac{u_\alpha}{R_1} - \frac{1}{B}\frac{\partial}{\partial\beta}\left(\frac{1}{B}\frac{\partial u_n}{\partial\beta}\right) - \frac{1}{A^2B}\frac{\partial B}{\partial\alpha}\frac{\partial u_n}{\partial\alpha} \tag{10}$$

$$\chi_{\alpha\beta} = \frac{1}{R_1}\frac{A}{B}\frac{\partial}{\partial\beta}\left(\frac{u_\alpha}{A}\right) + \frac{1}{R_2}\frac{B}{A}\frac{\partial}{\partial\alpha}\left(\frac{u_\beta}{B}\right) - \frac{1}{AB}\left(\frac{\partial^2 u_n}{\partial\alpha\partial\beta} - \frac{1}{A}\frac{\partial A}{\partial\beta}\frac{\partial u_n}{\partial\alpha} - \frac{1}{B}\frac{\partial B}{\partial\alpha}\frac{\partial u_n}{\partial\beta}\right) \tag{11}$$

where A and B are the lamé coefficients of the shell mid-surface; u_α, u_β and u_n are the displacement components in the curvilinear α, β and normal n directions (here n denotes the unit normal perpendicular to the shell mid-surface, i.e., the shell thickness direction), respectively; and R_1 and R_2 are the radius of curvature along the α and β curves, respectively, as shown in Fig.1.

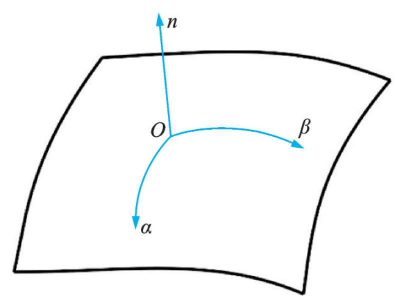

Fig.1 Curvilinear coordinate system of shell's mid-surface

1.2 Computational framework of PINN

In this section, we first introduce the basic theory of the classical artificial neural network. Then, we describe the construction of the energy-based loss function which constitutes the foundation of the proposed physics-informed neural network.

1.2.1 Fundamentals of artificial neural network

The basic architecture of ANN consists of three types of neuron layers: Input, hidden and output layers, as shown in Fig.2.

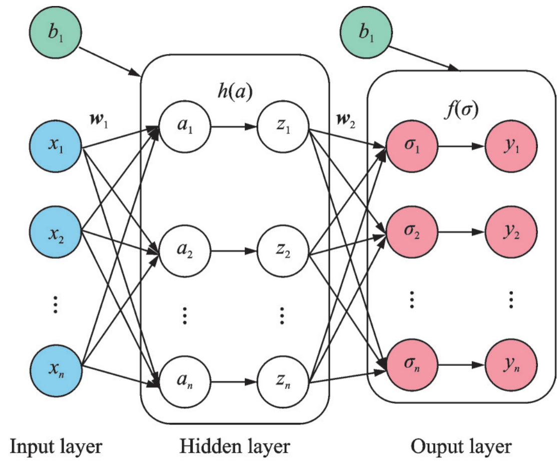

Fig.2 Architecture of a multi-layered neural network

In hidden layers, the output z_j^k of the jth neuron in the kth layer can be described as

$$z_j^k = h\left(\sum_{i=1}^{n} w_{ij}^k a_i^{k-1} + b_j^k\right) \tag{12}$$

where a_i^{k-1} is the output of the ith neuron in the $(k-1)$th layer, w_{ij}^k the weights assigned to the jth neuron in the kth layer that connects to the ith neuron in the $(k-1)$th layer, b_j^k the bias of the jth neuron in the kth layer, and $h(\cdot)$ the nonlinear activation function. It is noted that, the activation function does not apply to the output layer of the neural network.

The commonly used activation functions are: the logistic sigmoid function, the hyperbolic tangent function and the rectified linear unit function (ReLU)[8], etc. Both logistic sigmoid and hyperbolic tangent functions are nonlinear continuous which are able to approximate any target functions with arbitrary number of layers[19].

Considering a purely data-driven neural network, a loss function has to be defined to evaluate the fitting error of a neural network to a given data set. For thin-shell problems, the typical training data set can be defined as the displacement values at a set of locations. Then the mean square error (MSE) loss function of a regression problem can be defined as

$$L_D = \sum_{i=1}^{n_D} \frac{1}{n_D}\left[(u_\alpha^i - \hat{u}_\alpha^i)^2 + (u_\beta^i - \hat{u}_\beta^i)^2 + (u_n^i - \hat{u}_n^i)^2\right] \tag{13}$$

where n_D is the number of training samples, the superscript i the ith training sample, and "$\hat{}$" the observed values in the training data set.

Training is a process to optimize the weights and bias of the ANN such that the loss function is minimized. Classical optimization method is the stochastic gradient descent (SGD) method, in which the weights and bias of the ANN are updated as

$$\boldsymbol{w}^{(m+1)} = \boldsymbol{w}^{(m)} - \eta \frac{\partial L_D}{\partial \boldsymbol{w}} \tag{14}$$

$$\boldsymbol{b}^{(m+1)} = \boldsymbol{b}^{(m)} - \eta \frac{\partial L_D}{\partial \boldsymbol{b}} \tag{15}$$

where $\boldsymbol{w}$ and $\boldsymbol{b}$ are the matrix and vector form of the weights and bias, respectively, where the superscripts $m+1$ and m represent the iteration numbers, and η is a user-defined hyper-parameter called learning rate.

Alternatively, more computationally efficient methods, such as the Levenberg-Marquardt method and the Adam method[20], can be used to update the weights and bias of the ANN. The Adam algorithm is summarized as follows.

Algorithm 1 Adam method

(1) Initialize: $\boldsymbol{w}^{(0)}, \boldsymbol{b}^{(0)}$ ANN mapping $\boldsymbol{Y} = f(\boldsymbol{w}, \boldsymbol{b})$, the loss function $L = f(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{Y})$, constants α, β, η and ϵ

(2) Set $\boldsymbol{m}^{(0)}=0$, $\boldsymbol{v}^{(0)}=0$

(3) for $m=0, 1, 2, \cdots, n$ do

(4) Compute Jacobian matrix $\boldsymbol{J}^{(m)} = \partial L^{(m)}/\partial \boldsymbol{w}^{(m)}$

(5) Compute $\boldsymbol{m}^{(m+1)} = \gamma_1 \boldsymbol{m}^{(m)} + (1-\gamma_1)\boldsymbol{J}^{(m)}$

(6) Compute $\boldsymbol{v}^{(m+1)} = \gamma_2 \boldsymbol{v}^{(m)} + (1-\gamma_2)\boldsymbol{J}^{(m)} \odot \boldsymbol{J}^{(m)}$

(7) Compute $\hat{\boldsymbol{m}}^{(m+1)} = \dfrac{\boldsymbol{m}^{(m+1)}}{1-\alpha^{(m+1)}}$, $\hat{\boldsymbol{v}}^{(m+1)} = \dfrac{\boldsymbol{v}^{(m+1)}}{1-\beta^{(m+1)}}$

(8) Compute $\boldsymbol{w}^{(m+1)} = \boldsymbol{w}^{(m)} - \eta \dfrac{\hat{\boldsymbol{m}}^{(m+1)}}{\sqrt{\hat{\boldsymbol{v}}^{(m+1)}} + \epsilon}$

(9) Compute $L^{(m+1)}$

(10) if $L^{(m+1)} \leqslant$ tolerance then go to Step (14)

(11) else go to Step (3)

(12) end if

(13) end for

(14) Return optimized weights and biases $(\boldsymbol{w}^{(m+1)}, \boldsymbol{b}^{(m+1)})$

In Algorithm 1, the symbol $\odot$ represents the element-wise product of two matrices, γ_1 and γ_2 are the hyper-parameters that control the exponential decay rates for the moment estimates, and ϵ is a small value. Typically, these parameters take values of $\gamma_1=0.9$, $\gamma_2=0.999$, $\eta=0.001$ and $\epsilon = 10^{-8}$[15].

It is noted that, the derivatives of loss function with respect to the weights and bias in Eqs.(14,15) can be determined with the gradient back-propagation method[12].

1.2.2 Physics-informed neural network

Physics-informed neural network leverages the governing equations of a specific problem, thus may reduce the amount of data required for the training of the neural network. There exist several ways to construct the loss function of PINN for thin-shell problems, such as the PDE-based (strong form) and the energy-based (weak form) methods, as shown in Fig.3. In this paper, we mainly focus on the weak form approach, where the total potential energy of the shell is taken as the loss function

$$L_E = \Pi = U + V \tag{16}$$

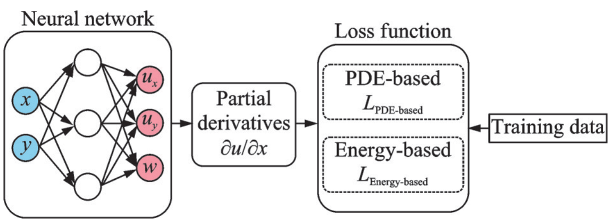

Fig.3 Flow chart of machine learning framework by different approaches

According to the principle of minimum potential energy, the minimization of the loss function Eq.(16) will result in the admissible displacements corresponding to the equilibrium configuration of the shell.

It is noted that, the essential boundary conditions should be considered during the construction of PINN. A simple way to impose these conditions is to modify the activation function of the output layer. For example, considering a simply supported beam with its length equal to L_a. Then the activation function $f(\sigma)$ of the output layer can be modified as

$$f(\sigma) = f(\sigma)x(x - L_a) \tag{17}$$

For more complex geometries and boundary conditions, such as clamped boundary conditions, more elaborated distance functions can be constructed. Alternatively, one can switch to the weak form by adding a penalty term in the loss function. For more details, we refer interested readers to Ref.[8].

The evaluation of the loss function Eq.(16) requires the numerical integration of the total potential energy, which can be approximated as

$$L_E \approx \sum_{i=1}^{n_s}(U_{\mathrm{m,e}}^i + U_{\mathrm{b,e}}^i)S_{\mathrm{e}}^i - \sum_{i=1}^{n_s}(\boldsymbol{F}\cdot\boldsymbol{u})^i S_{\mathrm{e}}^i - \sum_{i=1}^{n_b}(\boldsymbol{T}\cdot\boldsymbol{u})^i \Gamma_{\mathrm{e}}^i \tag{18}$$

where n_s and n_b are the number of sampling points of the shell's mid-surface and the Neumann boundary, respectively; $U_{\mathrm{m,e}}^i$ and $U_{\mathrm{b,e}}^i$ the membrane and bending strain energy density at the ith sampling point and can be obtained from Eqs.(2,3); $\boldsymbol{F}$ and $\boldsymbol{T}$ the volume force vector field and surface traction volume force; and S_{e}^i and Γ_{e}^i are the discrete area and length around the sampled point. Supposing the number of sampling points is sufficiently large, they can be approximated as $S_{\mathrm{e}}^i = S/n_s$, $\Gamma_{\mathrm{e}}^i = \Gamma/n_b$.

2 Examples

In this section, the accuracy and robustness of the proposed PINNs for the analysis of plate and shell structures are studied. Three numerical examples are presented, which are the plate under uniform pressure, the half-cylindrical shell and the relatively complex wing box structures. The prediction results of PINN will be compared to the FE results to validate the accuracy of the proposed PINN. It is noted that, in the following examples, we use random sampling strategies for the surface domain, and uniform sampling strategy for the boundary domain.

2.1 Deflection of a plate

In this example, a four-edge clamped square plate under uniform pressure is studied, as shown in Fig.4. The length and thickness of the plate are $L=20$ mm and $h=1$ mm, respectively. The Young's modulus of the plate is $E=70$ GPa, and the Poisson's ratio is $\mu=0.3$. The uniform pressure is set to be $p=0.1$ GPa.

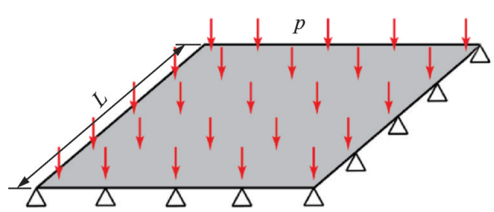

Fig.4 Plate deflection problem

To construct the PINN, the inputs are taken as the in-plane coordinates of the sampling points, and the outputs are taken as the in-plane and out-of-plane displacements u, v, w. The PINN consists of three different neural networks each representing a displacement component, as shown in Fig.5. Besides, five hidden layers and five neurons in each hidden layer are adopted in this example.

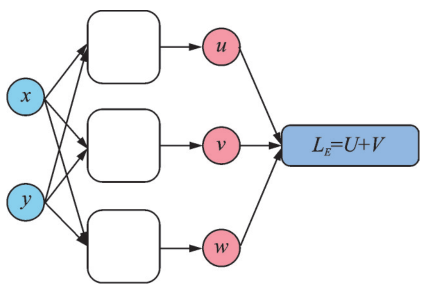

Fig.5 Deflection of a plate example: The basic structure of the PINN

The activation function of the PINN is chosen as the Tanh function. To incorporate the essential boundary conditions, the activation functions of the output layer are modified as

$$f_{u,v}(\sigma)=\sigma\cdot(x+10)(x-10)(y+10)(y-10) \tag{19}$$

$$f_w(\sigma)=\sigma\cdot\left[(x+10)(x-10)(y+10)(y-10)\right]^2 \tag{20}$$

where $f_{u,v}(\sigma)$ represent the activation functions of the outputs u and v components.

To train the PINN, a number of 10 000 samples are obtained and re-collected after every 200 training steps. The total number of training steps is set to be 10 000, and the learning rate is 10^{-3}. Figs.6(a) and (b) show the contour plots of the plate deflection obtained with PINN and FEM, respectively. Besides, a relative error between the two models is also shown in Fig.6(c). The deflection curves along middle line of the plate is shown in Fig.7. It is found that the proposed PINN provides results consistent with the FEM reference solution. The FEM solution is used as the reference solution for validation. The close agreement between the PINN predictions and FEM results demonstrates the effectiveness of the proposed method.

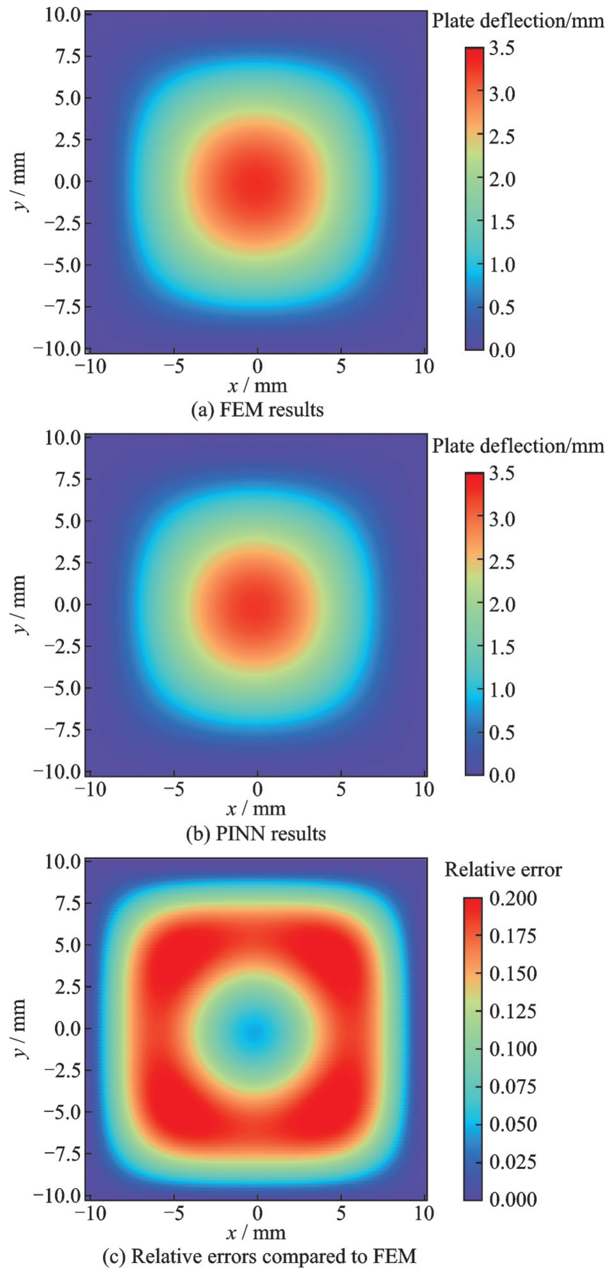

Fig.6 Comparisons of the plate deflection results

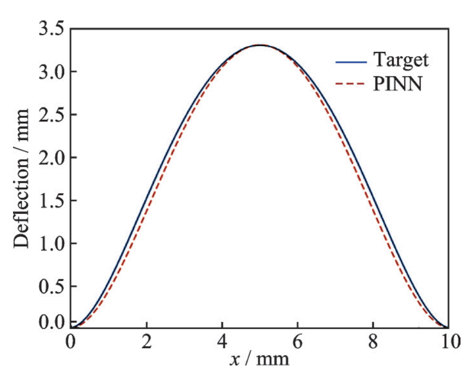

Fig.7 Comparisons of the deflection along the middle line of the plate

The accuracy of the prediction can be estimated by the coefficient of determination R^2, which is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N\sigma^2} \tag{21}$$

where N is the number of samples, y_i the prediction result, $\hat{y}_i$ the target result, and σ^2 the overall variance of the target. In this example, the value of R^2 is 0.993 6, which is very close to 1, showing that the deformation of the plate predicted by the proposed PINN is highly accurate.

2.2 Cylindrical shell example

In this example, a half-cylindrical shell subjected to a uniform pressure p=1 GPa is studied, as shown in Fig.8. The shell is clamped at the two straight boundaries and free at the two curved boundaries. The length and radius of the shell is L=20 mm and R=10 mm, respectively, and the shell thickness is h=1 mm. We set the Young's modulus and the Poisson's ratio the same as the first example.

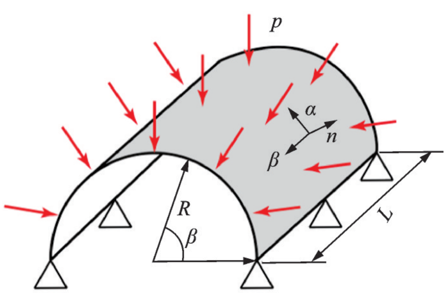

Fig.8 Half cylindrical shell example

The PINN consists of five hidden layers with five neurons in each hidden layer. To simplify the construction of the PINN, a cylindrical coordinate system is used. Thus, the number of inputs reduces to (θ, β), and the outputs are the three displacement components u_α, u_β, u_n.

In the cylindrical coordinate system, the boundary conditions can be formulated as

$$u_\alpha\big|_{\theta=0} = u_\beta\big|_{\theta=0} = u_n\big|_{\theta=0} = 0 \tag{22}$$

$$u_\alpha\big|_{\theta=\pi} = u_\beta\big|_{\theta=\pi} = u_n\big|_{\theta=\pi} = 0 \tag{23}$$

$$\left.\frac{\partial u_n}{\partial \theta}\right|_{\theta=0} = \left.\frac{\partial u_n}{\partial \theta}\right|_{\theta=\pi} = 0 \tag{24}$$

Then the activation functions of the output layer are reformulated as

$$f_{u_\alpha}(\sigma) = \sigma \cdot \theta(\theta - \pi) \tag{25}$$

$$f_{u_\beta}(\sigma) = \sigma \cdot \theta(\theta - \pi) \tag{26}$$

$$f_{u_n}(\sigma) = \sigma \cdot [\theta(\theta - \pi)]^2 \tag{27}$$

To train the constructed PINN, 10 000 sample points on the shell's mid-surface are adopted and updated after each 200 training steps. The total number of training steps is set to be 6 000. Fig.9 shows the displacement contour plots of the PINN and FEM model as well as their relative errors. It can be observed that, the energy-based PINN model predicts highly consistent results compared to the refer-

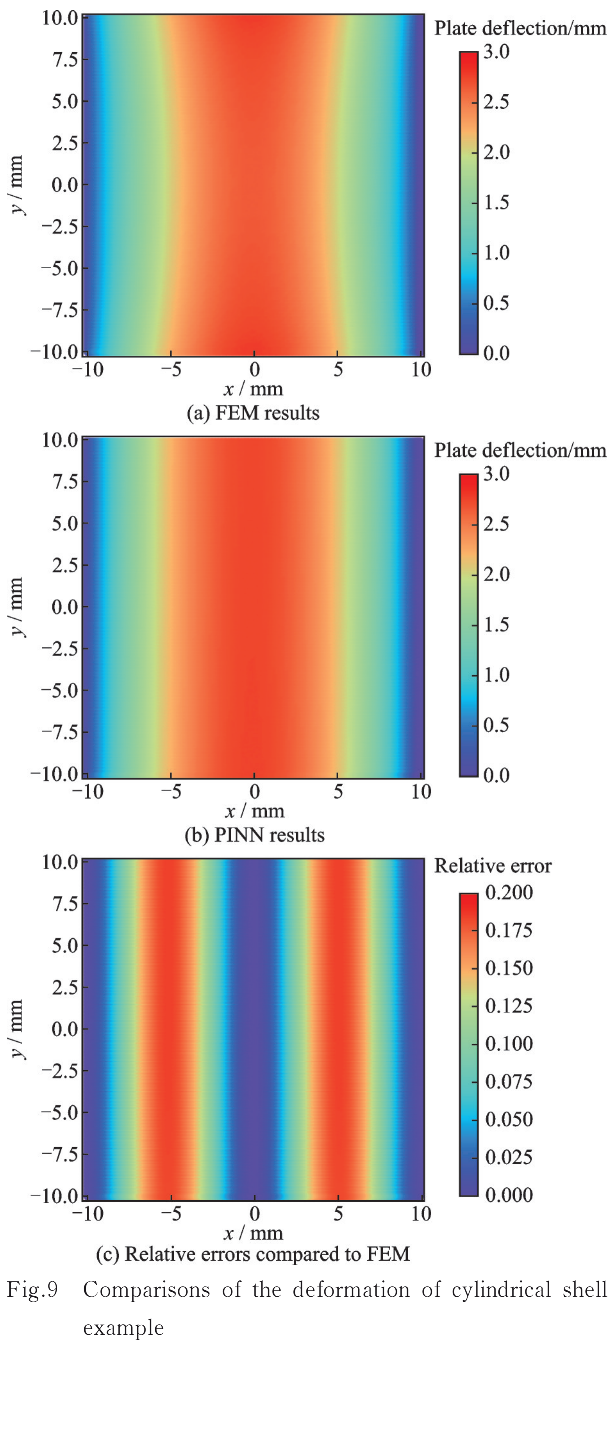

Fig.9 Comparisons of the deformation of cylindrical shell example

ence solutions. The displacements of the two models along the curve of $\beta=L/2$ are shown in Fig.10, which confirms the accuracy of the PINN. Besides, the value of R^2 for this example is 0.992 1 which is very close to 1.

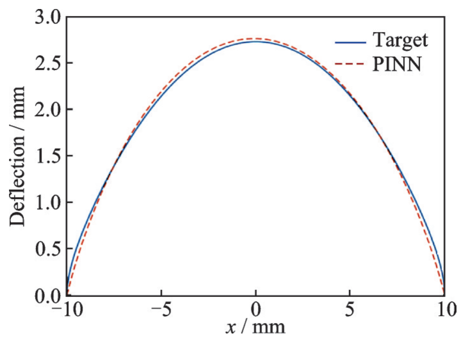

Fig.10 Comparisons of the displacement along the middle line of the cylindrical shell ($\beta=L/2$)

2.3 Wing box structure

In this example, a relatively complex example of a wing box structure is studied. The simplified wing structure consists of five patches and is clamped at one end, as shown in Fig.11. The geometric parameters and material properties of the wing box are shown in Table 1. We assume that the mean thickness of the wing is 80% of the maximum thickness. The upper surface of the wing box is subjected to a uniform distributed pressure of $p=$ 3 150 Pa.

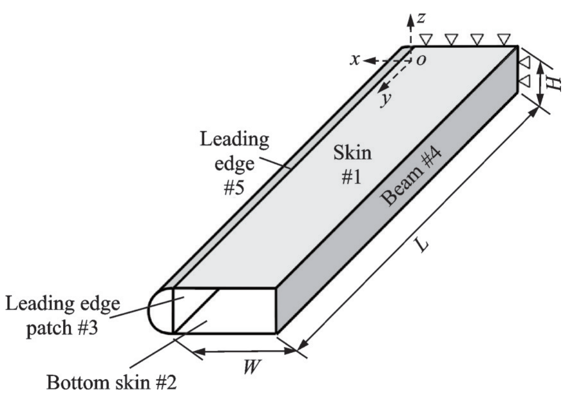

Fig.11 Wing box structure example

To facilitate the construction of PINN, we establish the local coordinate system for each patch, as shown in Fig.12. For each patch, an independent neural network is established, the inputs are taken as the local coordinates (α, β) and (θ, β), and the

Table 1 Geometric parameters and material properties of wing box

Parameter	Value
Maximum thickness of wing/mm	150
Thickness of the skin/mm	12
Thickness of the beam/mm	10
H/mm	120
L/mm	7 000
W/mm	600
Young's modulus E/GPa	70
Poisson's ratio μ	0.3

outputs are chosen as u_α, u_β, u_n. Then, these five neural networks are combined together to form the PINN of the wing box structure, as shown in Fig.13. It is noted that, five hidden layers with five neurons each layer are used in this example.

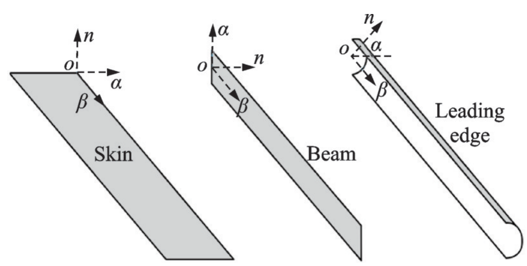

Fig.12 Local coordinate system of each patch of the wing box

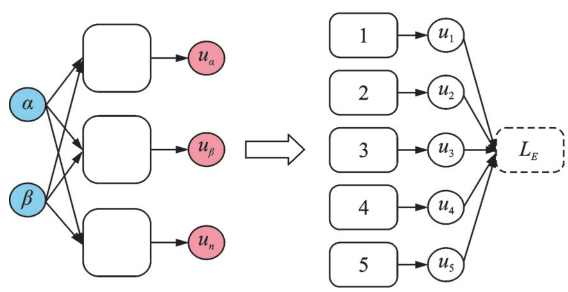

Fig.13 Wing box structure example: Schematic illustration of the PINN

In order to enforce the coupling constraints between patches, the loss function of the wing structure is extended as

$$L_E=\Pi=U+V+L_B+L_D \tag{28}$$

where L_B is the residuals of the constraints at the patch boundaries, shown as

$$L_B=\sum_{i=1}^{n_l}\frac{1}{N_{bl}^i}\sum_{j=1}^{N_{bl}^i}\lambda\left(\Delta u^{ij}\right)^2 \tag{29}$$

where n_l is the number of the coupling boundaries,

N_{bl}^{i} the number of training samples along the ith boundary, Δu^{ij} the difference of the displacements and rotations between the two coupled patches, and λ the penalty factor, which is taken as 1 in the reported examples.

In Eq.(28), L_D is MSE of the displacement at the coupling interfaces used to improve the prediction accuracy of the PINN, which is formulated as

$$L_D = \| \boldsymbol{u} - \hat{\boldsymbol{u}} \| \tag{30}$$

For the training of the PINN, 10 000 samples are used to compute the strain energy of each patch, and 2 000 samples are used for the evaluation of boundary constraints. For the evaluation of L_D, a relatively small number of sampling points of 160 are used. The learning rate of the PINN is set to be 10^{-3}, with a batch size of 200 and the total number of training steps of 6 000. In order to add the boundary conditions, the activation functions of the output layer are modified as

$$f_{u_\alpha}(\sigma) = \sigma \cdot \beta \tag{31}$$

$$f_{u_\beta}(\sigma) = \sigma \cdot \beta \tag{32}$$

$$f_{u_n}(\sigma) = \sigma \cdot \beta^2 \tag{33}$$

Fig.14 shows the comparisons of the wing box displacement contour plots obtained by FEM and energy-based PINN. Besides, the distributions of the displacement along the central line of the upper skin are shown in Fig.15. It can be found that, the proposed energy-based PINN predicts rather accurate results compared to the FEM reference solutions. In order to study the prediction accuracy of the PINN in detail, we also listed the R^2 values of each patch in Table 2. It is shown that, the results of patches 1 and 2 are more accurate than the rest of the patches which may due to the fact that, these two patches undertake larger part of the loads, thus have higher internal energy than the rest patches.

To further evaluate the computational efficiency of the proposed method, the computational cost of the PINN model is compared with that of the conventional FEM for the wing-box example. In the FEM approach, the main computational effort is associated with mesh generation and numerical solution of the governing equations. In contrast, the proposed PINN framework requires a training stage to optimize the neural network parameters through minimizing the physics-informed loss function.

In this study, the PINN model is trained using the Adam optimizer for several thousand iterations. Although the training stage requires additional computational time compared with a single FEM solution, once the training process is completed, the

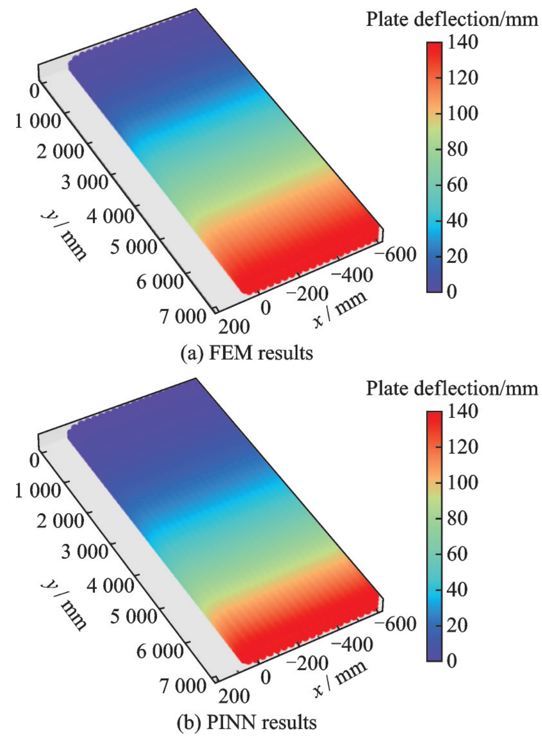

Fig.14 Comparisons of wing box displacements

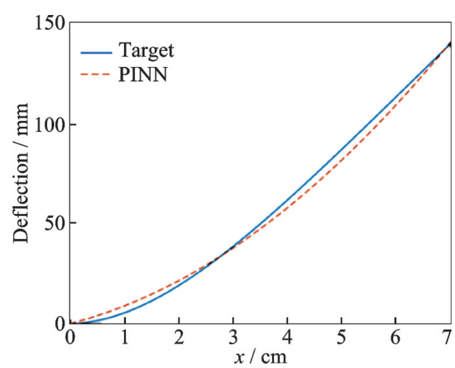

Fig.15 Comparisons of displacements along the central line of the wing box's upper skin

Table 2 R^2 of the wing box patches predicted by PINN

Patch	R^2
1	0.977 1
2	0.972 6
3	0.951 6
4	0.953 4
5	0.958 8

trained network can rapidly predict the structural response for new input parameters without solving the governing equations again. Therefore, the PINN model can serve as an efficient surrogate model for repeated evaluations and parametric studies of complex wing structures.

To provide a quantitative comparison, the computational time of both methods is summarized in Table 3, including the hardware configuration used in this study. The results indicate that while the PINN method requires additional training time, it provides fast prediction once the network is trained, which can be advantageous for applications involving repeated structural evaluations.

Table 3 Comparison of computational cost between PINN and FEM

Method	Hardwave	Computational stage	Time
FEM	CPU	Numerical solution	≈20 s
PINN	CPU	Training	≈12 min
PINN	CPU	Prediction	≈0.02 s

It can be observed that the FEM solution mainly involves mesh generation and numerical solution of the governing equations, which leads to a relatively short computation time for a single analysis. In contrast, the PINN framework requires a training stage to minimize the physics-informed loss function. Although the training process requires more computational time, the trained network can be directly used for rapid predictions without solving the governing equations again. Consequently, the proposed PINN framework can serve as an efficient surrogate model for repeated simulations or parametric studies of wing-box structures.

3 Summary and Outlooks

In this paper, a physics-informed neural network is proposed for the linear elastic static analysis of thin-shell structures. Compared to the traditional data-driven neural network, the proposed PINN uses total potential energy of the shell as the loss function, therefore, the training process of the PINN is equivalent to the minimization of the potential energy of the shell. Besides, in the proposed PINN, we use independent neural networks for each output variable, which provides the flexibility of adjusting the weights of contributions to the loss function.

The performance of the proposed PINN was illustrated with several numerical examples including the plate, the cylindrical shell as well as the relatively complex wing box structure. It can be found that, the proposed PINN does not require a large amount of training data which are sometimes difficult to obtain. The only inputs to the PINN are the coordinates of the sampling points, which significantly simplifies the construction and training process of the neural network.

For more complex wing box structure, we demonstrate the feasibility and accuracy of the proposed PINN in dealing with shell structures with multiple patches. The energy-based loss function can be extended with an extra term enforcing the patch compatibility constraints at the interfaces. Besides, we added a small amount of data-driven training data to the training process of the PINN, which results in a hybrid neural network and helps to improve the performance of the PINN.

All numerical results were compared with the FEM reference solutions, and good accuracy was achieved, demonstrating the effectiveness of the proposed machine learning framework. For the relatively complex wing-box structure, a direct next methodological step is to increase the network capacity by enlarging the network width and/or depth, followed by a systematic sensitivity study of the architecture, so as to clarify whether the remaining discrepancy mainly arises from underfitting due to insufficient model capacity rather than from the formulation itself. In addition, more advanced training strategies, such as using limited-memory broyden-fletcher-goldfarb-shanno (L-BFGS) after the initial Adam optimization and introducing adaptive loss weighting, can be considered to further improve convergence behavior and prediction accuracy. More complex structures combining beams, plates, and shells will also be studied, and more complex behaviors such as nonlinearities will be included.

References

[1] TISO P, JANSEN E, ABDALLA M. Reduction method for finite element nonlinear dynamic analysis of shells[J]. AIAA Journal, 2011, 49: 2259-2304.

[2] KOHAR C P, GREVE L, ELLER T K, et al. A machine leaning framework for accelerating the design process using CAE simulations: An application to finite element analysis in structural crashworthiness[J]. Computer Methods in Applied Mechanics and Engineering, 2021, 385: 114008.

[3] FAMILI A, SHEN W M, WEBER E, et al. Data preprocessing and intelligent data analysis[J]. Intelligent Data Analysis, 1997, 1(1): 3-23.

[4] LAGARIS I E, LIKAS A. Artificial neural networks for solving ordinary and partial differential equations[J]. IEEE Transactions on Neural Networks, 1998, 9(5): 987-1000.

[5] RAISSI M, PERDIKARIS P, KARNIADAKIS G E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations[J]. Journal of Computational Physics, 2019, 378: 686-707.

[6] E W N, YU B. The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems[J]. Communications in Mathematics and Statistics, 2018, 6: 1-12.

[7] 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[J]. Computer Methods in Applied Mechanics and Engineering, 2021, 383: 113933.

[8] BASTEK J H, KOCHMANN D M. Physics-informed neural networks for shell structures[J]. European Journal of Mechanics A—Solids, 2023, 99: 104960.

[9] YAN C A, VESCOVINI R, DOZIO L. A framework based on physics-informed neural networks and extreme learning for the analysis of composite structures[J]. Computers & Structures, 2022, 265: 106761.

[10] ZHUANG X, GUO H, ALAJLAN N, et al. Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning[J]. European Journal of Mechanics—A/Solids, 2021, 87: 104225.

[11] ZHANG X, GARIKIPATI K. Machine learning materials physics: Multi-resolution neural networks learn the free energy and nonlinear elastic response of evolving microstructures[J]. Computer Methods in Applied Mechanics and Engineering, 2020, 372: 113362.

[12] TAO F, LIU X, DU H, et al. Physics-informed artificial neural network approach for axial compression buckling analysis of thin-walled cylinder[J]. AIAA Journal, 2020, 58(6): 2737-2747.

[13] HUANG X J, WANG F J, ZHANG B R, et al. Enriched physics-informed neural networks for dynamic Poisson-Nernst-Planck systems[J]. Mathematics and Computers in Simulation, 2025, 237: 231-246.

[14] LI X, WANG F J, WANG R H, et al. Fundamental solution neural networks for solving inverse Cauchy problems for the Laplace and biharmonic equations[J]. Computers & Mathematics with Applications, 2026, 201: 1-17.

[15] WANG F J, LI X, LIU H Q, et al. An adaptive method of fundamental solutions using physics-informed neural networks[J]. Engineering Analysis with Boundary Elements, 2025, 178: 106295.

[16] JAGTAP A D, KHARAZMI E, KARNIADAKIS G E. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems[J]. Computer Methods in Applied Mechanics and Engineering, 2020, 365: 113028.

[17] JAGTAP A D, KAWAGUCHI K, KARNIADAKIS G E. Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations[J]. Communications in Computational Physics, 2020, 28(5): 2002-2041.

[18] LEISSA A. Vibration of shells: NASA SP-288[R]. [S.l.]: NASA, 1973.

[19] CYBENKO G. Approximation by superpositions of a sigmoidal function[J]. Mathematics of Control, Signals and Systems, 1989, 2: 303-314.

[20] KINGMA D P, BA J L. ADAM: A method for stochastic optimization[C]//Proceedings of International Conference on Learning Representations. New York, USA: [s.n.], 2015: 1-15.

Acknowledgement This work was supported by the National Natural Science Foundation of China (No.12472202).

Authors

The first author Mr. GUO Boyu is currently pursuing the M.S. degree at Nanjing University of Aeronautics and Astronautics. His research interests include structural optimization and computational mechanics.

The corresponding author Dr. **GUO Yujie** works as an associate professor in College of Astronautics at Nanjing University of Aeronautics and Astronautics. His research interests include computational mechanics, isogeometric analysis, structural stability and dynamics, structural optimization, composite and smart structures, and digital twins for aircraft systems.

Author contributions Mr. **GUO Boyu** designed the overall study, developed the energy-based PINN framework, implemented the loss functions, carried out the numerical simulations, analyzed the results, and drafted the manuscript. Dr. **GUO Yujie** proposed the research topic, supervised the study, guided the methodology, checked the theoretical derivations, and revised the manuscript. Mr. **XU Xianhua**, Mr. **CHEN Yuhang**, and Mr. **QIU Hao** contributed to the implementation and testing of the numerical examples, including FEM reference modeling, parameter studies, and post-processing of the results. They also assisted in the literature review, preparation of figures and tables, and proofreading of the manuscript. All authors discussed the results, contributed to the interpretation of the findings, commented on the manuscript, and approved the final version.

Competing interests The authors declare no competing interests.

(Production Editor: ZHANG Huangqun)

基于能量的物理信息神经网络在多片板壳结构线弹性分析中的应用

郭博宇[1]，陈宇航[2]，邱　浩[1]，徐先华[1]，郭玉杰[1]

(1.南京航空航天大学航天学院，南京 211106，中国；2.中国空气动力研究与发展中心，绵阳 621000，中国)

摘要：机器学习为物理模型预测提供了一种快速而准确的技术手段。针对板壳结构线弹性静力变形分析问题，本文建立了一种基于物理信息神经网络(Physics-informed neural network，PINN)的机器学习框架。与纯数据驱动神经网络不同，该方法将物理规律引入网络训练过程，从而有效降低了对训练数据的依赖。本文以薄壁结构总势能泛函构造PINN的损失函数，并通过在损失函数中引入界面协调约束，将所提方法推广至多片板壳结构分析。同时，针对不同板壳结构算例，对基于能量损失函数的PINN预测性能进行了评估，并与有限元结果进行了对比。数值结果表明，本文所提出的框架能够在较少训练数据条件下获得较高精度的预测结果，体现出其在板壳结构线弹性分析中的有效性与适用性。

关键词：物理信息神经网络；机器学习；板壳结构；总势能；偏微分方程

研究亮点：

1. 提出了一种基于总势能损失函数的物理信息神经网络框架，用于薄壁板壳结构线弹性静力分析。
2. 通过在损失函数中引入界面协调约束，实现了所提方法向多片板壳结构分析的扩展。
3. 方板、圆柱壳及多片翼盒算例验证了该方法在较少训练数据条件下仍具有较高预测精度。
4. 所提方法为复杂板壳结构的高效、物理一致性智能分析提供了新的计算思路。