Finite Elements in Analysis and Design 216 (2023) 103893

Contents lists available at ScienceDirect

Finite Elements in Analysis & Design

journal homepage: www.elsevier.com/locate/finel

A robust physics-informed neural network approach for predicting structural instability

Hau T. Mai [a,b], Tam T. Truong [c], Joowon Kang [d], Dai D. Mai [e], Jaehong Lee [a,*]

[a] Deep Learning Architectural Research Center, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, Republic of Korea
[b] Faculty of Mechanical Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Viet Nam
[c] Institute for Computational Science and Artificial Intelligence, Van Lang University, Ho Chi Minh City, Viet Nam
[d] School of Architecture, Yeungnam University, 280, Daehak-Ro, Gyeongsan, Gyeongbuk 38541, Republic of Korea
[e] Faculty of Mechanical Engineering, Ho Chi Minh City University of Technology and Education, Ho Chi Minh City, Viet Nam

ARTICLE INFO

Keywords:
Neural networks
Critical points
Geometric nonlinear
Structural stability
Direct physics-informed neural network

ABSTRACT

In this work, a direct physics-informed neural network (DPINN) is first proposed to analyze the stability of truss structures that incremental-iterative algorithm is completely removed from the implementation process. Instead of resolving of nonlinear equations as in conventional numerical methods, a neural network (NN) is employed to minimize the loss function which is designed to guide the training network based on the structural instability information. In our computational framework, the parameters including weights and biases of the network are considered as design variables. In addition, spatial coordinates of joints are examined as input data, while corresponding displacements and load factor unknown to the network are taken account of output. To address this challenge, the predicted outputs obtained by feedforward are utilized to establish the loss function relied on the residual load and stiffness characteristics of the structure as the first stage. And then, back-propagation and optimizer are applied to automatically calculate sensitivity and adjust parameters of the network, respectively. This entire process known as training is repeated until convergence. To that end, the position of the critical point is indicated as soon as the training ends by our network without using any time-consuming incremental-iterative algorithms as well as structural analyses. Several benchmark examples of truss structures associated with the geometric nonlinearity influence are investigated to evaluate the efficiency of the proposed scheme. The obtained results reveal that the present framework is extremely simple to implement and also yields the strong robustness as well as higher accuracy.

1. Introduction

In fact, nonlinear behaviors become significant in most of the structures, especially for the lightweight, slender, and complex geometry designs such as bars, arches, shells, plates, etc [1,2]. Hence, they must also be taken into account when analyzing the structural failure. Therein, nonlinear stability analysis is an integral part and plays a vital role in the structural design process [3,4]. It has received much attention in the computational mechanics community. To investigate the instability, critical points must be identified along the equilibrium path, which is a major challenge for finite element analysis (FEA) related to the singular solution [1].

In general, many different algorithms were developed by researchers for solving structural stability problems. And therein, they were commonly classified into two main groups called indirect and direct methods. The first one relies on a detecting parameter that will be evaluated during each incremental load step to gain equilibrium [5–7]. For instance, Riks [8] tried to locate the limit point by using Newton's method. Simo et al. [9] introduced a scaling vector to control the constraints for tracking limit points. In order to save the computational effort, Shi and Crisfield [10] developed a semi-direct approach to identify singular points. Besides, an exact technique for the determination of critical points was delivered by Chan [11]. And more recently, several alternatives have been developed to estimate the singular points [12–14]. In spite of its success, some variants still face challenges, such as increased computational cost due to intermediate solutions and its dependency on incremental-iterative techniques as well as control parameters. To circumvent these drawbacks, the direct approach has been introduced, in which a system of nonlinear equations is derived to directly detect the critical points. During the last decades, it has received much interest from researchers and achieved

* Corresponding author.
E-mail addresses: maitienhaunx@gmail.com (H.T. Mai), truongtam0303@gmail.com (T.T. Truong), kangj@ynu.ac.kr (J. Kang), daimd@hcmute.edu.vn (D.D. Mai), jhlee@sejong.ac.kr (J. Lee).

https://doi.org/10.1016/j.finel.2022.103893
Received 27 October 2022; Received in revised form 6 December 2022; Accepted 9 December 2022

the remarkable success in this field [15–18]. Wriggers et al. [19] were among the first authors to develop this method to find the singular points. A critical displacement framework is developed for predicting structural instability by Oñate and Matias [20]. Also, Battini et al. [21] introduced an improved minimal augmentation approach for the elastic stability of beam structures. Despite their remarkable success in the stability analysis, they still have certain limitations. Firstly, as indicated by Shi [1], it required a good starting vector, sufficiently close to the singular point, and gradient information to obtain convergence. Moreover, the criterion for selecting the starting point highly depends on expert experience, as well as the incremental-iterative methods.

In recent years, machine learning (ML) has led to outstanding achievements in computational mechanics such as materials science [22,23], structural analysis [24,25], structural optimization [26–30], fluid mechanics [31], structural health monitoring [32–36], and so on. One of the machine ML models called the neural network have been widely used in science and engineering [37]. Therein, physics-informed neural network (PINN) has emerged as a new, powerful, and efficient approach to solve problems in mechanics, and received increasing attention in the engineering community in recent times. In contrast to the data-driven approach, it incorporates governing physical laws as well as given design data (e.g. boundary conditions (BCs), geometry, properties of materials, etc.) into the training process of the model with unknown solutions [24]. Indeed, PINN has successfully solved complex problems, such as elasticity [38], plasticity [39,40], hyperelasticity [41,42], fluid [31,43], material [44,45], thermal [46,47], and so on. However, it has still not been yet utilized for structural stability analysis thus far.

To overcome the above-discussed challenge, a robust physics-informed NN framework is firstly proposed in this study to directly locate the first critical point of truss structures without using any incremental-iterative algorithms. In place of solving the structural stability problems as conventional methods, NN is designed to directly locate the critical point by minimizing the loss function. Therein, the weights and biases of the network are regarded as design variables. And the geometry information containing coordinate values of joints is considered as the input data which are easily gathered. In specific, it is worth emphasizing that this approach relies on unsupervised learning, so it only requires the input data and does not need corresponding output data. In this work, the output values of the network are the unknown displacement field and load factor which are expressed by parameters of the network and the input values, respectively. And the training process aims to seek them that the loss function is minimized. To achieve such an objective, the predicted outputs of the network obtained from the feedforward (FF), instability information, and BCs are employed to design the loss function which includes the unbalanced load and test function. The back-propagation (BP) and optimizer are then used to automatically compute gradient and update network parameters. This process is repeated until convergence. And finally, the position of the critical point corresponding to the NN's optimal parameters is found instantly at the training process ends without using any incremental-iterative techniques. The efficiency and applicability of the proposed model are also demonstrated through several numerical examples to analyze the stability of truss structures. The obtained results showed that DPINN not only simple in procedure, but also yields higher accuracy.

The remainder of this work is structured as follows. Section 2 presents a brief outline of the stability analysis. Next, Section 3 describes a robust direct physics-informed NN framework. In Section 4, several truss structures are tested to show the accuracy, effectiveness, and robustness of our approach. Finally, Section 5 outlines some of the most important conclusions.

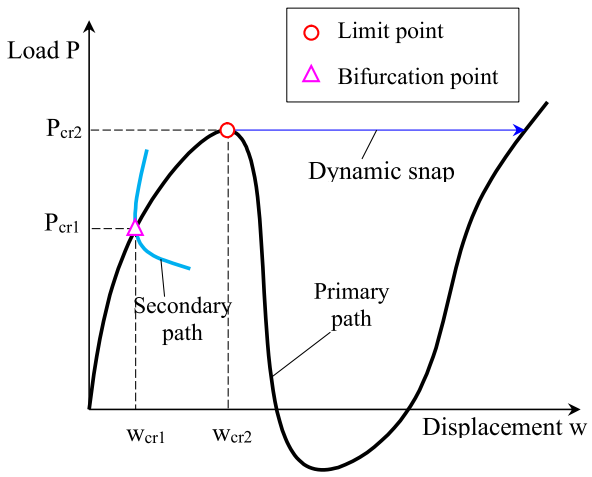

Fig. 1. Schematic representation of snap-through and bifurcation points.

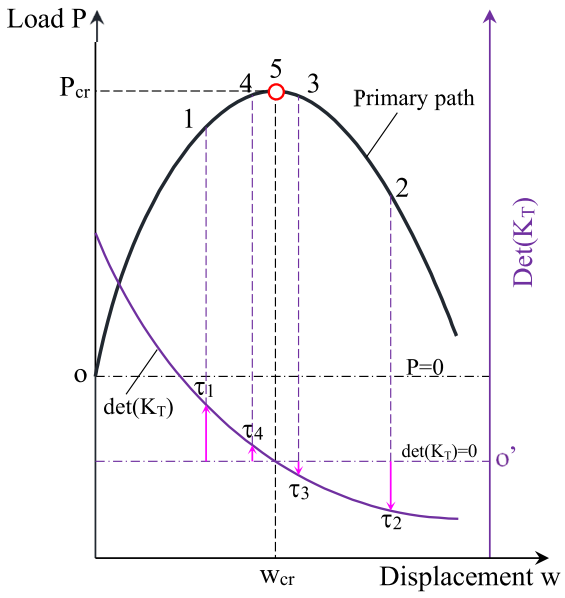

Fig. 2. Indirect approach using bi-section method.

2. Elastic structural stability

The primary objective of structural stability analysis is to determine the position of critical points. And their outstanding characteristic is a singular tangent stiffness matrix $\mathbf{K}_T$ [12]. When the structural system may lose its stability in dealing with various geometrical or material properties or both changes. However, this work only considers the effect of geometrical changes on the instability arising. Based on the phenomena of structural instability, there are two types of singular points: limit (snap-through) and bifurcation points, as shown in Fig. 1. Note that the sign of the determinant of the stiffness matrix $\mathbf{K}_T$ changes in the vicinity of the limit point, whereas it does not vary with the bifurcation point [48].

For the indirect method, its underlying principle relies on the combination of a test function and numerical methods to estimate the singular point [49,50]. And this test function allows evaluating the positive definiteness of $\mathbf{K}_T$ such as the determinant, smallest pivot, smallest eigenvalue, etc. This scheme is illustrated in Fig. 2 where Bisection method is applied to locate the singular point (5) based on the sign change of the test function τ.

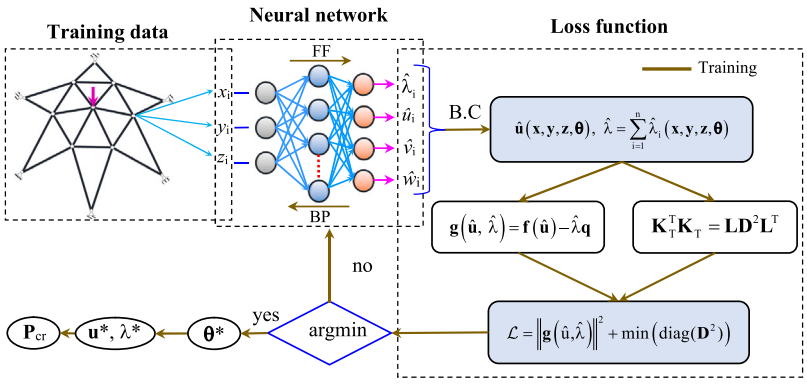

Fig. 3. The whole process of the direct instability-informed neural network framework.

In the second tactic, the direct procedure, which is known as one of the most powerful techniques, was developed and widely applied in structural stability analysis to enhance computational efficiency [15–17]. As noted in this procedure, the critical points are indicated directly by solving a system of nonlinear equations consisting of the equilibrium and test functions [51]. The interested reader is referred to [2] for more details about deriving them. It is worth mentioning that the iterative method is used only to obtain the singular point, and there would not trace the behavior of the structure for the direct methods.

3. Direct physics-informed neural network framework

A direct physics-informed NN framework is first presented in this section to detect the point of structural instability. Herein, the NN acts as an identification model to detect the critical point by minimizing the loss function, and its whole process is shown in Fig. 3. According to the flow diagram, the initial parameters of the network were set to zeros, which corresponds to the unloading state of the structure. The coordinates of each joint are treated as a training data point, while unknown displacements and load factors are referred to as output data. As a consequence of these outputs, loads, and BCs, the loss function involving the residual load and property of the stiffness matrix is established to minimize by the training process. To achieve the goal, FF, BP, and optimizer are employed to adjust the parameters. As soon as the training process completed, the first critical point will be located by our model corresponding to the minimum loss function. To understand clearly our approach, the following sub-sections represent in greater detail the input data, training NN, and designing loss function.

3.1. Training data

It should be noted that all training data of DPINN contains only a collection of coordinates (x_i, y_i, z_i) and BCs of all structural joints. Obviously, it is easily obtained from the geometric information of the structure. Furthermore, for planar and space structures, its size is small ($m \times 2$) and ($m \times 3$), respectively. Here m is the number of joints, while 2 or 3 denotes the number of spatial coordinates for a two-dimensional or three-dimensional truss. In addition, the displacements and load factors are unknown quantities, and not contained in the training data. Alternatively, they are the outputs of the network, which need to be estimated to minimize the loss function. Once trained, the proposed network is easily able to identify the critical point corresponding to the outputs.

3.2. Neural network

As one of the ML models, the NN has been widely used to assist in decision-making for various problems. It mimics the human brain activity that the relationship between the outputs and inputs is found by training. For illustration, a fully connected NN with three layers as a baseline architecture is depicted in Fig. 4. Therein, the first layer is known as the input layer which includes three units with respect to the coordinates (x, y, z). The second layer is referred to as a hidden layer with m_h units. Herein, the number of hidden units deals with the complexity of the problems. And the output layer is the final layer with four neurons corresponding the predicted displacements and load factor $(\hat{\lambda}, \hat{u}, \hat{v}, \hat{w})$. All units of the previous layer are connected to all neurons in the present layer via the weights $\mathbf{W}^{(l)}$ and biases $\mathbf{b}^{(l)}$, which are called the parameters of the NN.

The training process aims at adjusting these parameters of the network to minimize the loss function. According to achieve this goal, two algorithms including FF and BP are utilized and they will repeat over and over until a convergence criterion is satisfied. Firstly, the FF is applied for mapping data from the input to output neurons which is expressed as $\hat{\mathbf{I}}$: $\mathbb{R}^3 \rightarrow \mathbb{R}^4$. And then, the governing relation between system inputs and outputs of each layer is expressed as

$$\begin{aligned} &\text{input layer} \quad : \quad \hat{\mathbf{o}}_0 = [x, y, z] \ \in \mathbb{R}^3, \\ &\text{hidden layer} : \quad \hat{\mathbf{o}}_1 = f_1\left(\mathbf{W}^{(1)}\hat{\mathbf{o}}^0 + \mathbf{b}^{(1)}\right) \in \mathbb{R}^{m_h}, \\ &\text{output layer} \ : \quad \hat{\mathbf{o}}_2 = f_2\left(\mathbf{W}^{(2)}\hat{\mathbf{o}}^1 + \mathbf{b}^{(2)}\right) = \left[\hat{\lambda}, \hat{u}, \hat{v}, \hat{w}\right] \ \in \mathbb{R}^4, \end{aligned} \tag{1}$$

where $f(.)$ denotes the activation function. Its role is to add non-linearity to the network by deciding whether a neuron should be activated or not, which makes the network capable of learning about the complete relationship between the inputs and outputs. There are some common choices, such as ReLU, LeakyReLU, Linear, Sigmoid, Softmax, Tanh, and so on. $\hat{\mathbf{o}}_1$ and $\hat{\mathbf{o}}_2$ are in turn the output of the hidden and output layers.

Based on the output values in the first phase, a loss function $\mathcal{L}$ is built to minimize. Next, BP is applied to achieve the sensitivity of the loss concerning the parameters of the network. As a consequence, their values are adjusted in the direction of the gradient descent, as represented below

$$\begin{aligned} {}^{t+1}W_{pq}^{(l)} &= {}^{t}W_{pq}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial {}^{t}W_{pq}^{(l)}}, \\ {}^{t+1}b_{p}^{(l)} &= {}^{t}b_{p}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial {}^{t}b_{p}^{(l)}} \qquad for \ \ 1 \le l \le 2, \end{aligned} \tag{2}$$

where η denotes the learning rate; p and q refer to the number of neurons in the lth and $(l-1)^{th}$ layer, respectively; t is the number of iterations. Finally, the loss function will converge to the minimum value after t iterations.

3.3. Loss function

Obviously, we strictly define the residual load vector $\mathbf{g}\left(\hat{\mathbf{u}}, \hat{\lambda}\right)$ and tangent stiffness matrix $\mathbf{K}_T$ with the obtained displacement field and load factor by FF. In order to achieve the strain, a space truss element

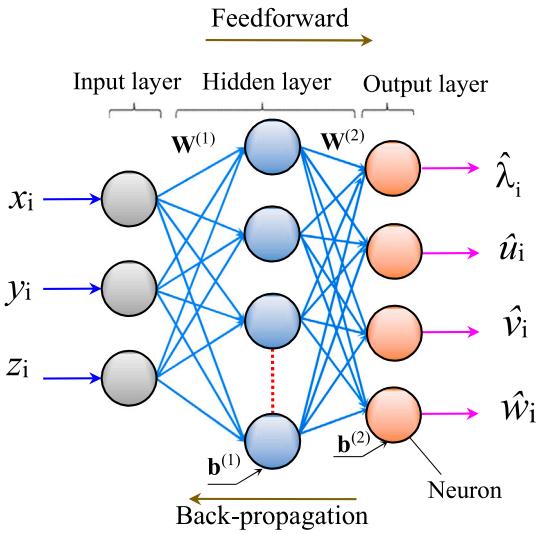

Fig. 4. A fully connected neural network.

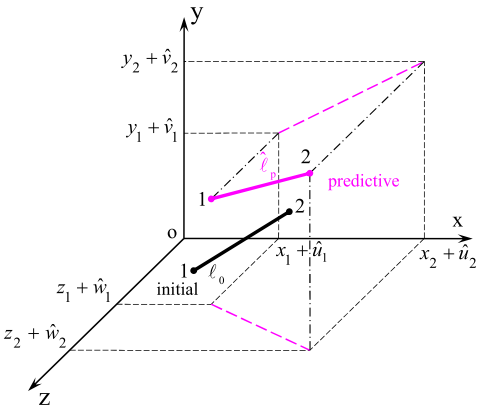

Fig. 5. Deformation of a space truss member.

in the initial and predictive configurations is considered as shown in Fig. 5. The coordinates of the member $(x_1,\ y_1,\ z_1,\ x_2,\ y_2,\ z_2)$ present the original configuration. Hence, the initial length ℓ_0 of the element is defined as follows

$$\ell_0 = \sqrt{\ell_{0x}^2 + \ell_{0y}^2 + \ell_{0z}^2}, \tag{3}$$

where $\ell_{0x} = x_2 - x_1$; $\ell_{0y} = y_2 - y_1$; $\ell_{0z} = z_2 - z_1$.

Let $(\hat{u}_1,\ \hat{v}_1,\ \hat{w}_1,\ \hat{u}_2,\ \hat{v}_2,\ \hat{w}_2)$ correspond to predicted displacements of the predictive configuration. Thus, the predictive length $\hat{\ell}_p$ of the element is defined as follows

$$\hat{\ell}_p = \sqrt{\hat{\ell}_{px}^2 + \hat{\ell}_{py}^2 + \hat{\ell}_{pz}^2}, \tag{4}$$

in which $\hat{\ell}_{px} = \ell_{0x} + \hat{u}_2 - \hat{u}_1$; $\hat{\ell}_{py} = \ell_{0y} + \hat{v}_2 - \hat{v}_1$ and $\hat{\ell}_{pz} = \ell_{0z} + \hat{w}_2 - \hat{w}_1$.

It can be seen that the axial strain is easily obtained from ℓ_0 and $\hat{\ell}_p$. According to Crisfield et al. [52], Green's strain is defined as

$$\varepsilon = \frac{\hat{\ell}_p^2 - \ell_0^2}{2\ell_0^2}. \tag{5}$$

Once the strain is found, the internal force vector of the kth member is given by

$$\mathbf{f}_k = E_k A_k \varepsilon_k \ell_0^{(k)} \mathbf{b}_k, \tag{6}$$

where E and A are the elastic modulus and the cross-section area, respectively; the vector $\mathbf{b}$ is defined as follows

$$\mathbf{b}_k^T = \frac{1}{\ell_0^{(k)}} \begin{bmatrix} -\hat{\ell}_{px}^{(k)} & -\hat{\ell}_{py}^{(k)} & -\hat{\ell}_{pz}^{(k)} & \hat{\ell}_{px}^{(k)} & \hat{\ell}_{py}^{(k)} & \hat{\ell}_{pz}^{(k)} \end{bmatrix}. \tag{7}$$

Hence, the residual load vector $\mathbf{g}(.)$ is strictly determined by the global internal force $\mathbf{f}$, external forces $\mathbf{q}$, and load factor $\hat{\lambda}$, which is expressed by

$$\mathbf{g}\left(\hat{\mathbf{u}},\ \hat{\lambda}\right) = \mathbf{f}\left(\hat{\mathbf{u}}\right) - \hat{\lambda}\mathbf{q}, \tag{8}$$

where $\hat{\mathbf{u}}(\mathbf{x}, \mathbf{y}, \mathbf{z}, \boldsymbol{\theta})$ is the displacement field with the fulfilled boundary conditions; $\boldsymbol{\theta}$ denotes the parameter vector of the network including weights and biases; $\mathbf{x}, \mathbf{y}$, and $\mathbf{z}$ are the coordinate vectors of the joints; $\hat{\lambda}$ is computed by a sum of the predicted load factor components λ_i, as follows

$$\hat{\lambda} = \sum_{i=1}^{n} \hat{\lambda}_i\ (x, y, z, \theta). \tag{9}$$

The tangent stiffness matrix of the kth element is obtained by differentiating the internal force vector with respect to the nodal displacements, which is given by

$$\mathbf{K}_T^{(k)} = E_k A_k \ell_0^{(k)} \mathbf{b}_k \mathbf{b}_k^T + \frac{E_k A_k \varepsilon_k}{\ell_0^{(k)}} \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 \\ \mathbf{I}_3 & \mathbf{I}_3 \end{bmatrix}, \tag{10}$$

in which $\mathbf{I}_3$ is the identity matrix of order 3. Obviously, the global stiffness matrix $\mathbf{K}_T$ is easily derived by assembling the element stiffness matrices.

According to indicate the critical point, the loss function of the DPINN is designed relying on a sum of the residual load and test function. And its mathematical formulation is expressed as follows

$$\mathcal{L} = \left\| \mathbf{g}\left(\hat{\mathbf{u}},\ \hat{\lambda}\right) \right\|^2 + \min\left(diag(\mathbf{D}^2)\right), \tag{11}$$

where $\mathbf{D}$ is a diagonal matrix obtained by singular value decomposition (SVD) of the $\mathbf{K}_T$, which is given by

$$\mathbf{K}_T = \mathbf{L}\,\mathbf{D}\,\mathbf{L}^T. \tag{12}$$

In this study, the lowest entry of the main diagonal of $\mathbf{D}$ is selected as a test function, so the eigenvalues of $\mathbf{K}_T$ and the main diagonal of $\mathbf{D}$ are the same roles. Furthermore, it allows to reducing computational costs for solving the problem [12,53,54]. Note that this work is implemented in the library of PyTorch with Python language. Hence, the computation of the SVD of $\mathbf{K}_T$ becomes easily dealt with by utilizing a LINALG tool which is availably integrated into the PyTorch library.

Instead of resolving the system of nonlinear equations consisting of the residual load and the test function as the conventional direct method [2], its residual as a loss function is minimized by our approach to indicate the optimal parameters $\boldsymbol{\theta}^*$ of the network.

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \left(\mathcal{L}\left(\boldsymbol{\theta}\right) \right). \tag{13}$$

It should be noted that the complex sensitivity analysis regarding computing the derivatives of the loss function with respect to each of all parameters can be easily dealt with by utilizing an automatic differentiation tool which is availably integrated into the PyTorch library. Once the network is trained, the critical point is found at the minimum of the loss function corresponding to the optimal parameters without using any incremental-iterative methods.

4. Numerical experiments

In the following, several well-known benchmark problems are investigated to demonstrate the simplicity, efficiency, and robustness of the proposed procedure for the stability analysis of truss structures. The found results using DPINN compare with results of existing conventional approaches in the literature and the indirect approach combined with Bisection method (Indirect-BM). Note that the total Lagrangian

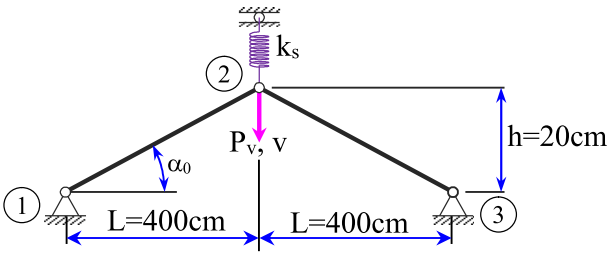

Fig. 6. Two bar planar truss.

kinematic description was employed to build the loss function of the DPINN for all numerical examples. As mentioned in Section 3.1, the number of input and output neurons are fixed for each problem, and it is easy to estimate. Therein, the number of neurons in each hidden layers is set equal for all the network. Additionally, the developed shallow networks consist of 1 to 2 hidden layers with no more than 50 neurons in each hidden layer. In order to the optimal architecture of the network, the trial-error and grid search methods were applied to find out the number of hidden layers as well as hidden neurons. In addition to perform the training, Adam [55] with a learning rate of 0.01 was utilized as an optimizer in this work. Besides, the linear transfer and hyperbolic tangent activation functions were employed for the output and hidden layers, respectively. It should be noted that all initial parameters of the network including weights and biases are set to zeros. Finally, the relative error in the L^2 norm is used as one of the criteria to evaluate the accuracy of DPINN.

4.1. Two-bar planar truss

A simple two-bar planar truss, as illustrated in Fig. 6, is examined as the first numerical example for the stability analysis. The cross-sectional area and Young's modulus of all truss members are set as A = 6.53 cm^2 and E = 20500 kN/cm^2, respectively. A concentrated load P_v is applied at the center, and its maximum value is 50 kN. To reach the goal, a three-layer network (2-10-3) is established for finding the critical point with 1,000 epochs. The exact solution of this problem was given by Pecknold et al. [56], where the stability of structure related to various geometrical and stiffness k_s of the spring. And the nondimensional constant K_u presents the combined nonlinear effects, as expressed below

$$K_u = \frac{k_s h}{AE\sin^3(\alpha_0)}. \tag{14}$$

In particular, if K_u is greater than or equal to 0 and less than 1, there exist two distinct limit points on the primary path, and their position is given by

$$\begin{cases} P_{cr_{1,2}} = AE\sin^3(\alpha_0)\left[K_u \pm 2\left(\frac{1-K_u}{3}\right)^{3/2}\right], \\ v_{cr_{1,2}} = h\left[1 \mp \sqrt{\frac{1-K_u}{3}}\right]. \end{cases} \tag{15}$$

When K_u is set equal to 1, a unique snap-through exists at

$$\begin{cases} P_{cr} = AE\sin^3(\alpha_0), \\ v_{cr} = h. \end{cases} \tag{16}$$

And there is no snap-through phenomenon for other K_u values. In this example, the effective performance of the DPINN is evaluated with several values of K_u such as 0, 0.25, 0.5, 0.75, and 1.

Firstly, this structure is considered for the case $K_u = 0$. As can be seen from Table 1, the analysis results obtained from the DPINN, including the first limit point, relative error, eigenvalue, residual, as well as minimum loss function, are a good agreement with the exact

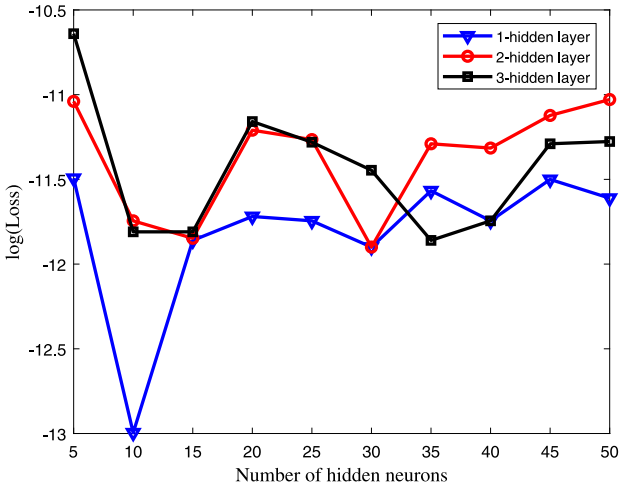

Fig. 7. Logarithm of loss function with varying hidden neurons and layers.

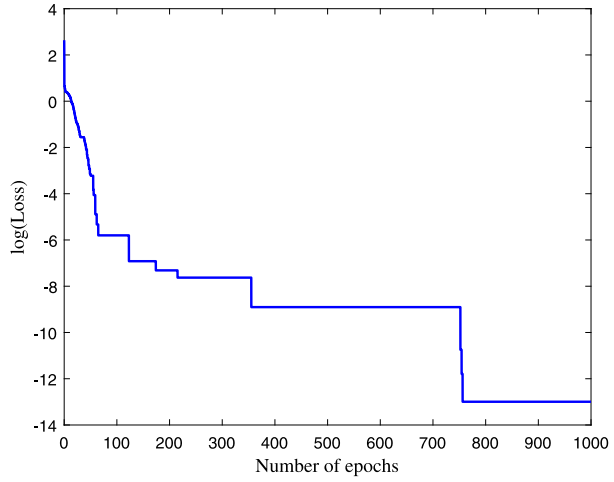

Fig. 8. The loss convergence history of the two-bar truss with $K_u = 0$.

solution for all strain measures. It is easily seen that the relative errors are smaller than 0.2%. Obviously, Green's strain is the best measure with the smallest error (0.0784%). Additionally, its loss function value including the eigenvalue of stiffness matrix and residual load vector is closest to zero among all four measures. Thus, Green's strain is utilized throughout the present work.

Based on the above architecture, a survey is adopted to investigate the influence of optimizer and activation function on the accuracy of the model. Accordingly, Table 2 reports the relative errors of the exact and the predicted results. From the data in this table, it is easily seen that Adam and Tanh are the best optimizer and activation function, respectively. On the contrary, Adadelta and Softmax are the worst choices. Obviously, the combination of Adam and Tanh gives the lowest relative error of 0.078%, so it is chosen for the training phase. At the same time, the grid search procedure is applied to optimize the number of hidden neurons and layers. The minimum loss function values are shown in Fig. 7 for each case. According to the obtained results, it can be observed that increasing the number of hidden layers or hidden neurons does not always improve the accuracy of the model. And it is plain that the network architecture with a single hidden layer and ten neurons is the most suitable for this example with the smallest loss function (1.01×10^{-13}).

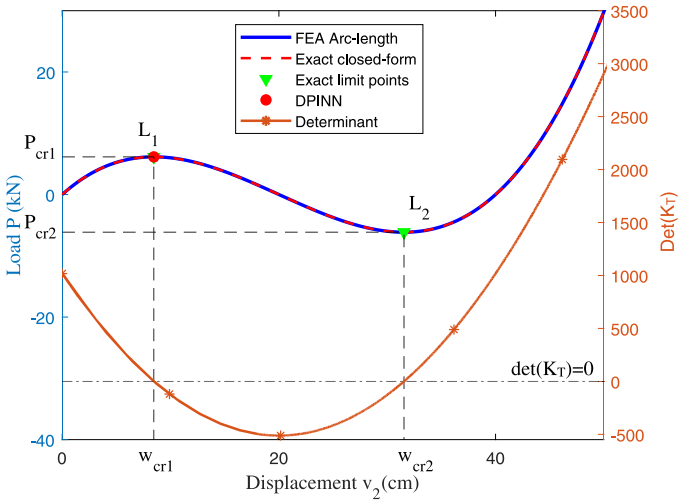

Fig. 9. Load–deflection curve for the two-bar truss with $K_u = 0$.

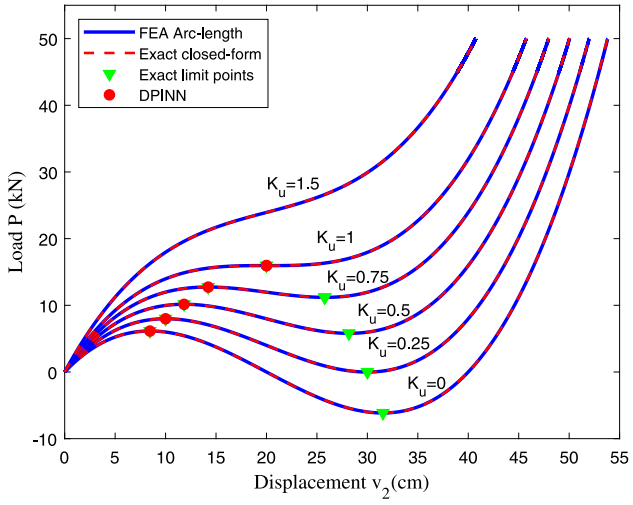

Fig. 10. Effect of the spring stiffness on the structural stability.

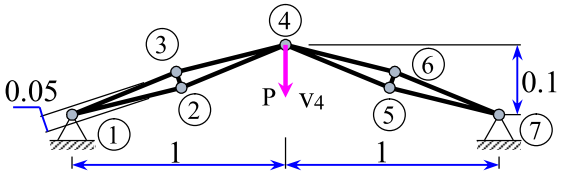

Fig. 11. Simple arch-truss structure.

Table 1
Comparison results obtained for the two-bar truss in searching the first snap-through point with $K_u = 0$.

Results	Exact	DPINN			
	Solution [56]	Engineering	Green	Log	Almansi
v_{cr_1} (cm)	8.4530	8.4577	8.4594	8.4593	8.4626
P_{cr_1} (kN)	6.1414	6.1492	6.1465	6.1517	6.1567
Relative error (%)	–	0.0873	**0.0784**	0.1162	0.1734
Eigenvalue (10^{-6})	–	0.4055	**0.2375**	0.4511	0.4493
Residual (10^{-6})	–	0.7851	**0.2117**	0.9070	2.0714
Loss (10^{-12})	–	0.7808	**0.1012**	1.0262	4.4927

Table 2
Relative error % of the first limit point with various activation functions and optimizers for the case $K_u = 0$.

Activation	Optimizers							
function	Adam	RMSprop	Rprop	Adagrad	Adamax	Adamw	SGD	Adadelta
Softmax	0.092	9.996	0.909	24.602	3.639	1.006	60.280	51.920
Sigmoid	0.087	0.680	0.278	23.491	7.170	1.968	42.092	13.673
Softplus	0.095	0.202	0.230	2.544	1.155	1.526	2.781	16.397
ReLU	0.087	0.151	0.117	1.494	0.408	0.643	0.751	1.618
Softsign	0.087	0.093	0.120	0.283	0.110	0.195	0.382	0.576
Tanh	**0.078**	0.088	0.099	0.089	0.097	0.122	0.199	0.112

The convergence curve depicted in Fig. 8 describes a detailed performance view of the proposed procedure. From the displayed result, the convergence rate at the first 100 epochs is very large and the optimal solution reaches after 800 epochs. On the other hand, the equilibrium path and the limit points are illustrated in Fig. 9. Firstly, it can easily be seen that the load–deflection curve at node 2 obtained with the FEA using arc-length was a good agreement with the exact closed-form solution. It follows readily from this result that the mechanical characteristics of geometric nonlinearity, including the residual load vector and stiffness matrix, as well as the arc-length method are enough reliability to solve the nonlinear structure problem. As shown in Fig. 9, there were two distinct limit points where the tangent stiffness matrix determinants are zeros. Clearly, it was a strong agreement among the first critical point L_1 obtained by DPINN (6.1465 kN) to the exact solution (6.1414 kN) with a very small error (0.083%).

To evaluate the robust performance of DPINN, the structural stability due to varying K_u will also be carefully considered. Accordingly, the location of the first critical points obtained by DPINN and the exact closed-form are mentioned in Table 3 and shown in Fig. 10. As observed, the results found by the NN are very close and agree well with the exact solution. More concretely, the relative errors are smaller than 0.08% for the nondimensional spring constants of 0, 0.25, 0.5, and 0.75, while this value is 0.38% at K_u =1. It is easily explained that the fidelity susceptibility at the first critical point and its vicinity for the case K_u =1 does not exhibits a sharp peak as in other cases. And this point looks like a saddle point of the optimization problem. It often poses a serious challenge to the conventional approaches in achieving the goal, so it is very sensitive to the parameters of the numerical method. It is clear that DPINN shows its effectiveness and accuracy for locating the first critical point with a relative error smaller than 0.4% when the stability change.

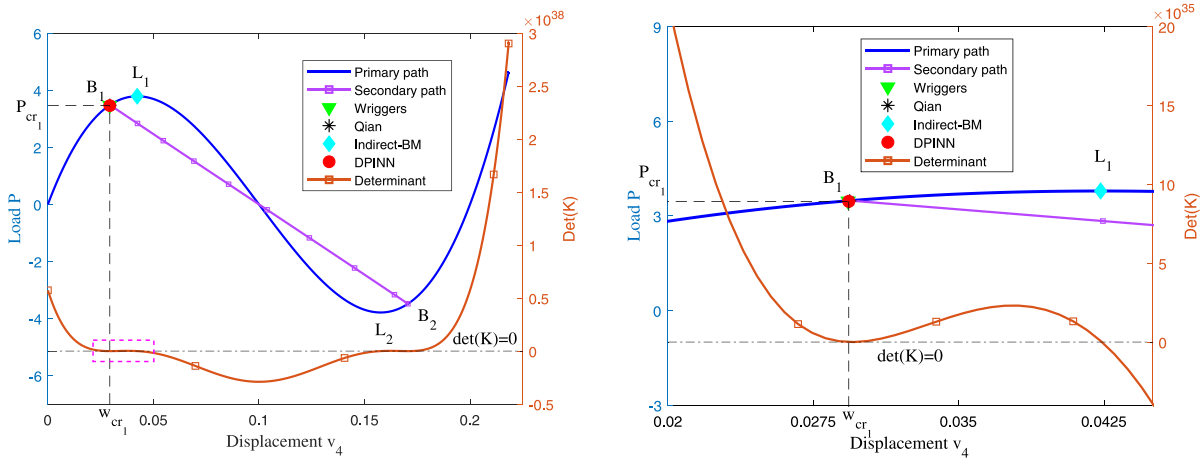

Fig. 12. Structural response of the simple arch truss: load-deflection curve and critical points.

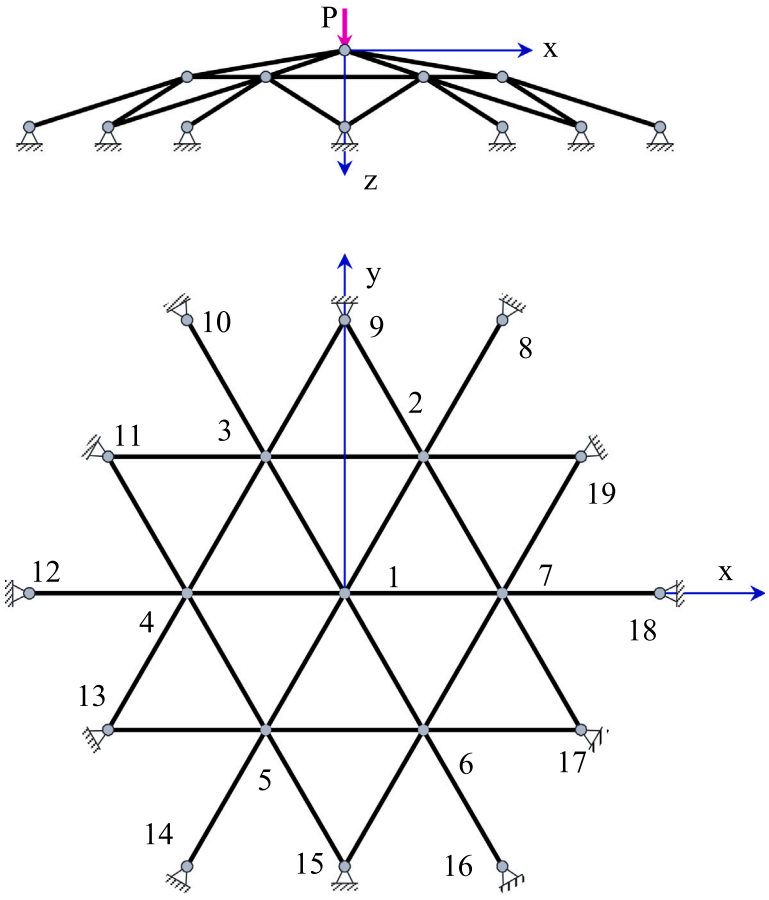

Fig. 13. Star dome structure with 30 members.

Table 3
Comparison results obtained for the two-bar truss in searching the first snap-through point with varying K_u.

Results	Exact Solution [56]					DPINN				
	$K_u = 0$	$K_u = 0.25$	$K_u = 0.5$	$K_u = 0.75$	$K_u = 1$	$K_u = 0$	$K_u = 0.25$	$K_u = 0.5$	$K_u = 0.75$	$K_u = 1$
v_{cr_1} (cm)	8.453	10.000	11.835	14.226	20.000	8.459	10.005	11.836	14.221	20.097
P_{cr_1}	6.141	7.978	10.149	12.734	15.956	6.146	7.983	10.155	12.739	15.956
Relative error (%)	–	–	–	–	–	0.078	0.057	0.037	0.038	0.380
Eigenvalue (10^{-6})	–	–	–	–	–	0.238	0.522	0.745	1.967	5.960
Residual (10^{-6})	–	–	–	–	–	0.212	1.426	2.630	0.542	8.381
Loss (10^{-12})	–	–	–	–	–	0.101	2.305	7.474	4.163	105.762

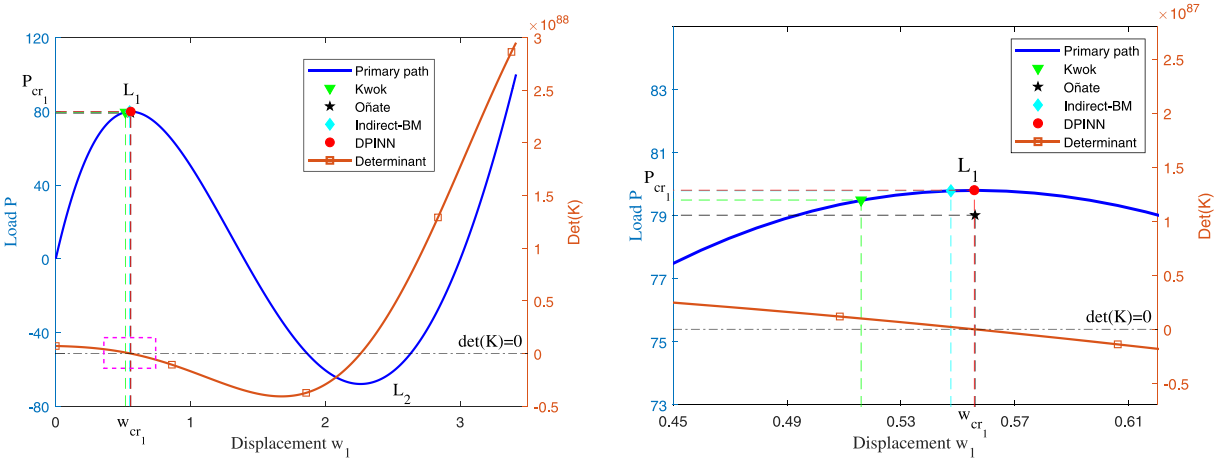

Fig. 14. Structural response of the star dome truss: load-deflection curve and critical points.

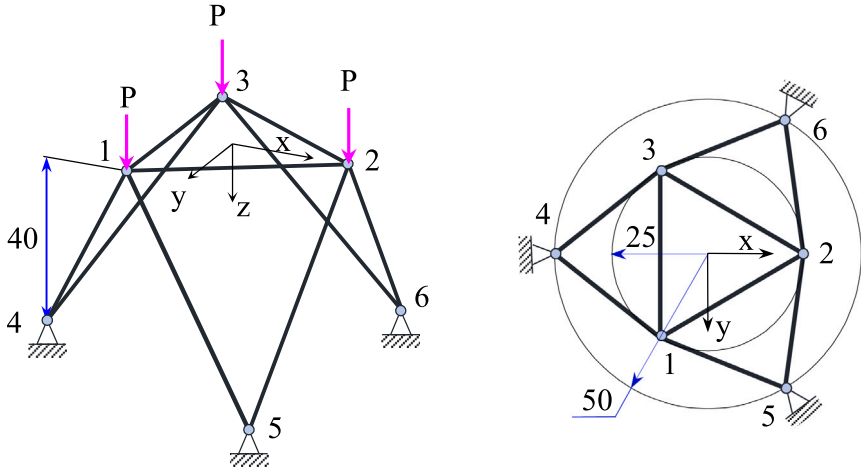

Fig. 15. Triangular truss dome.

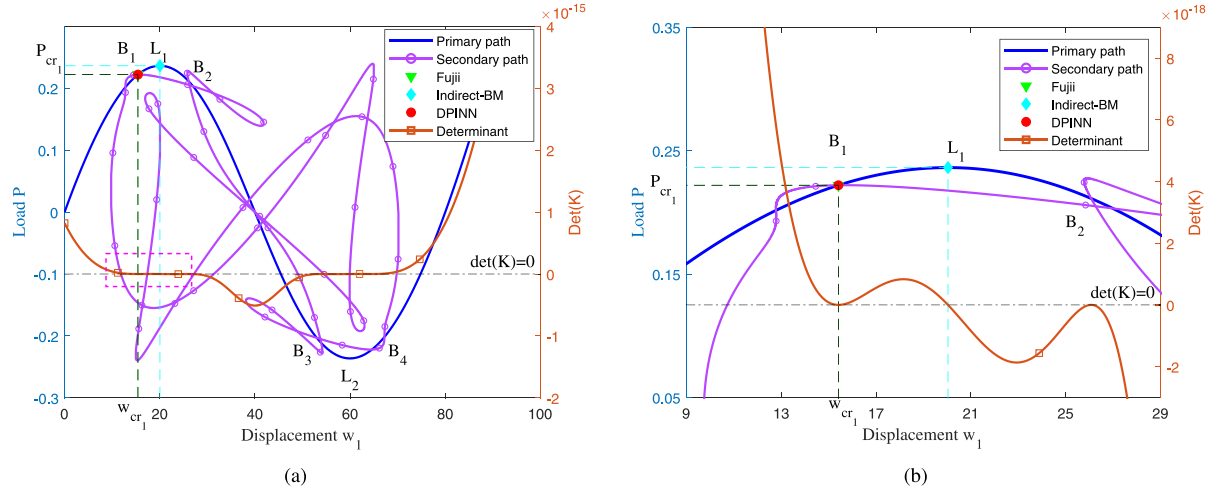

Fig. 16. Load-deflection curve and critical points of the triangular truss dome.

4.2. Simple arch-truss structure

A simple arch-truss structure is investigated as the second example for the stability analysis, as shown in Fig. 11. The geometry, loading, and properties of the material are dimensionless, and all members are the same with EA = 5000. The vertical force P is applied at the top node with the allowable maximum value of 5. This structure is widely used by researchers, such as Wriggers [19], Oñate [57], and

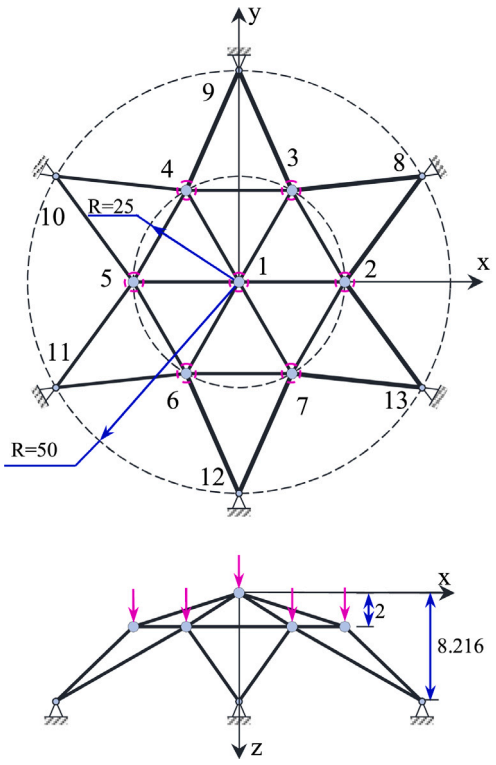

Fig. 17. Star dome truss with 24 bars.

Qian [58] for evaluating the performance of the algorithms. In this case, the network architecture, including two hidden layers and 15 neurons in each hidden layer, is selected for the training process with 1,000 epochs.

As previously proposed, the obtained stability analysis results by DPINN compared with other studies are shown in Table 4 and Fig. 12. Therein, the magenta dashed box at the lower left of Fig. 12a is magnified to Fig. 12b. It can be seen from the response curves that this structure exhibits two stability modes due to the bifurcation and limit points. As shown in Fig. 12b, the bifurcation point B_1 is the first critical point that occurs before the reached snap-through point L_1. As expected, the bifurcation point is found by DPINN (P_{cr_1} =3.4612), which is a good agreement with the previous studies. Although the relative error gained by Qian [58] (0.1153%) is slightly lighter than the DPINN (0.2548%), it is small and negligible. While the indirect approach using Bisection method only found the limit point L_1. This can easily be explained by the determinant does not change its sign in the vicinity of the bifurcation B_1, so several indirect methods are impossible to interpolate with the determinant [1]. Clearly, the DPINN outperforms the indirect-BM although it requires more computational times. Furthermore, the traditional approaches, including direct and indirect tactics, usually depend on the initial value selection, as well as control parameters of the incremental-iterative procedures [1,12,21]. In the converse direction, our approach allows us to identify the instability point without using any incremental-iterative techniques. In addition, the initial state without external loads is fixed as a starting point in all experiments. Hence, the DPINN is significantly easier to implement than conventional methods.

4.3. 30-bar dome truss

Next, the stability analysis of the 30-bar dome space truss is investigated as the third example, which has been previously introduced by Kwok [59] and Oñate [20]. The finite element representation and coordinates of nodes are shown and listed in Fig. 13 and Table 5,

Table 4
Comparison results obtained for the simple arch-truss structure in searching the bifurcation point.

Results	Wriggers [19]	Oñate [57]	Qian [58]	Indirect-BM	DPINN
w_{cr_1}	0.0293	–	0.0294	0.0423	0.0293
P_{cr_1}	3.4700	3.4700	3.4740	3.7895	3.4612
Relative error (%)	–	–	0.1153	9.2151	0.2548
Times (s)	–	–	–	4.6863	45.8369

Table 5
Coordinates of the node points of star dome structure.

Node	X	Y	Z
1	0	0	0
3	−15	25.981	1.5
4	−30	0	1.5
9	0	60	6
10	−30	51.962	6
11	−51.96	30	6
12	−60	0	6

Table 6
Comparison results obtained for the 30-bar dome truss in searching the first critical point.

Results	Kwok [59]	Oñate [20]	Indirect-BM	DPINN
w_{cr_1}	0.5160	0.5560	0.5475	0.5557
P_{cr_1}	79.5000	79.0200	79.7927	79.8129
Relative error (%)	–	0.6059	0.3704	0.3967
Times (s)	–	–	9.1752	63.8789

respectively. The vertical z-direction force P is applied at the apex of the star dome with the allowable maximum value of 100. The structural properties of all members are the same, dimensionless, and are equal to EA = 10^6. In addition, a NN (3-10-4) with 1,000 epochs is the finest performance for this application.

Table 6 and Fig. 14 provide a comparison between the result of the DPINN and other available algorithms in the literature for the first snap-through point. As can be seen in Fig. 14b, DPINN outperforms the other two studies. Clearly, the position of the first limit point obtained by the NN lies on the primary path, making the determinant of the stiffness matrix zero, while the other works can only approximate one of the two conditions mentioned above, as shown in Fig. 14(b). It is easily explained that the approaches of Kwok [59], Oñate [20] and the indirect-BM employed the incremental-iterative procedure. Consequently, it is very sensitive to the control parameters and often produces errors at large curvature positions [1,12,21]. Although the computational time of DPINN (63.8789 s) is larger than the indirect-BM (9.1752 s), it also yields higher accuracy.

4.4. Regular-triangular truss dome

Next, a regular-triangular truss dome is illustrated in Fig. 15. The material properties of all members are considered the same EA = 1, with the consistence unit. A concentrated force P in the positive direction of the z-axis is applied on the free nodes. As in the previous example, this structure contains two stability modes that deal with the bifurcation and limit points, respectively. A NN architecture with one hidden layer, 45 nodes per layer, and 1,000 epochs is designed for the training process.

Likewise, the stability analysis results achieved by this work are summarized in Table 7 for a comparison with another study. And a very good agreement is easily found between DPINN and Fujii [54] with a very small relative error of 1.62×10^{-4}. While the indirect-BM failed to indicate the first critical point B_1. Once again shows the present work demonstrates its robustness and effectiveness in detecting the singular point. The equilibrium paths of the structure at node 3 are plotted in Fig. 16. Here two different types of critical points exist

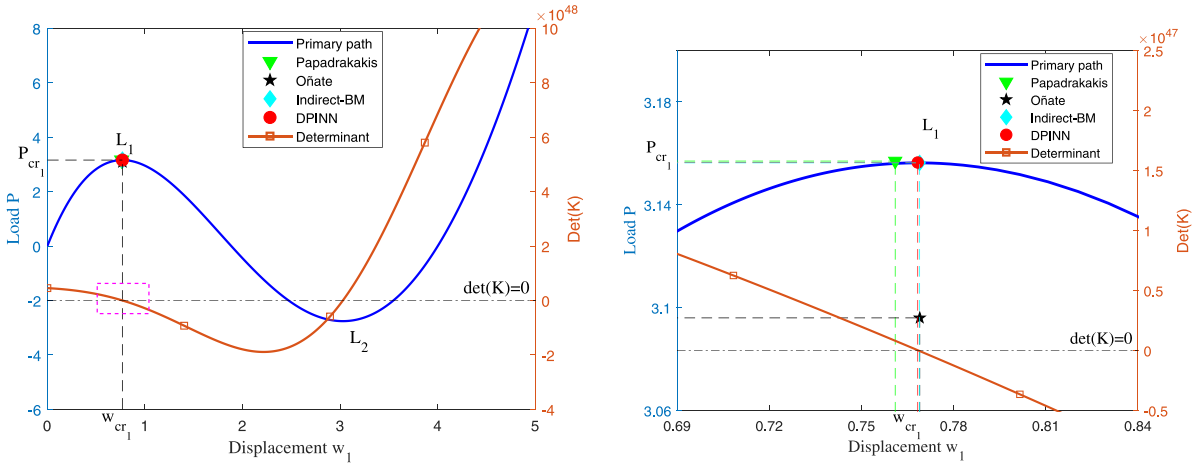

Fig. 18. Case 01: Load-deflection curve and critical points of the 24-bar dome truss.

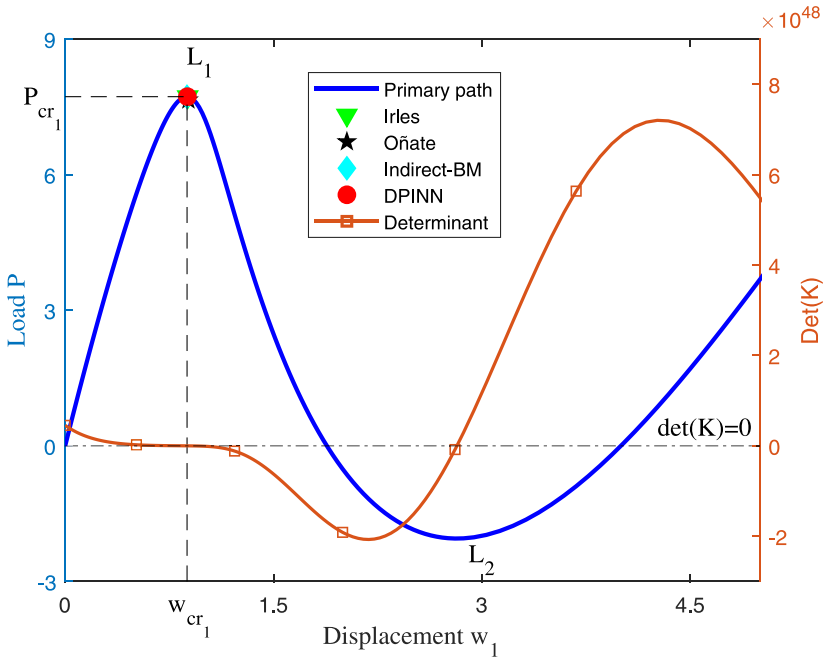

Fig. 19. Case 02: Load-deflection curve and critical points of the 24-bar dome truss.

Table 7
Comparison of results of the triangular truss dome for searching the bifurcation point.

Results	Fujii [54]	Indirect-BM	DPINN
w_{cr_1}	15.4131	20.0361	15.41309
P_{cr_1}	0.2219	0.2363	0.22195
Relative error (%)	–	29.9907	0.0003
Times (s)	–	12.1417	59.2046

in this structure, one for the bifurcation points (B_1, B_2, B_3, B_4), the other one for the limit points (L_1, L_2). And along the primary path, the first bifurcation point B_1 occurs first and then L_1, B_2 points follow in sequence. Hence, the first buckling mode is found at $P_{cr_1} = 0.22195$ for the instability of this structure. As the previously presented example, the current approach requires the higher computational cost (59.2046 s) compared to the indirect-BM (12.1417 s), but the DPINN shows its efficiency in significantly improving the accuracy.

4.5. Star dome truss

A star dome truss containing 24 members shown in Fig. 17 is considered as the last example. All members are assigned the same property of material EA = 10,000 with consistence unit. It is subjected to concentrated forces in the positive direction of the z-axis. Three loading cases given in Table 8 are considered. It has been examined previously by Papadrakakis [60], Oñate [20], Irles [61], and Crisfeld [52]. To identify the critical point, the network configuration with one hidden layer, 15 hidden neurons, and 1,000 epochs is chosen to perform the stability analysis.

Table 9 compares the results obtained by the present approach and other methods. Next, a detailed discussion of each case is proposed. The limit point causes the loss of stability of the structure in the first case, as shown in Fig. 18. It is clear from the data in Table 9 and Fig. 18b that the obtained result of the DPINN reveals a fairly good agreement with the indirect-BM and outperforms the other algorithms. More concretely,

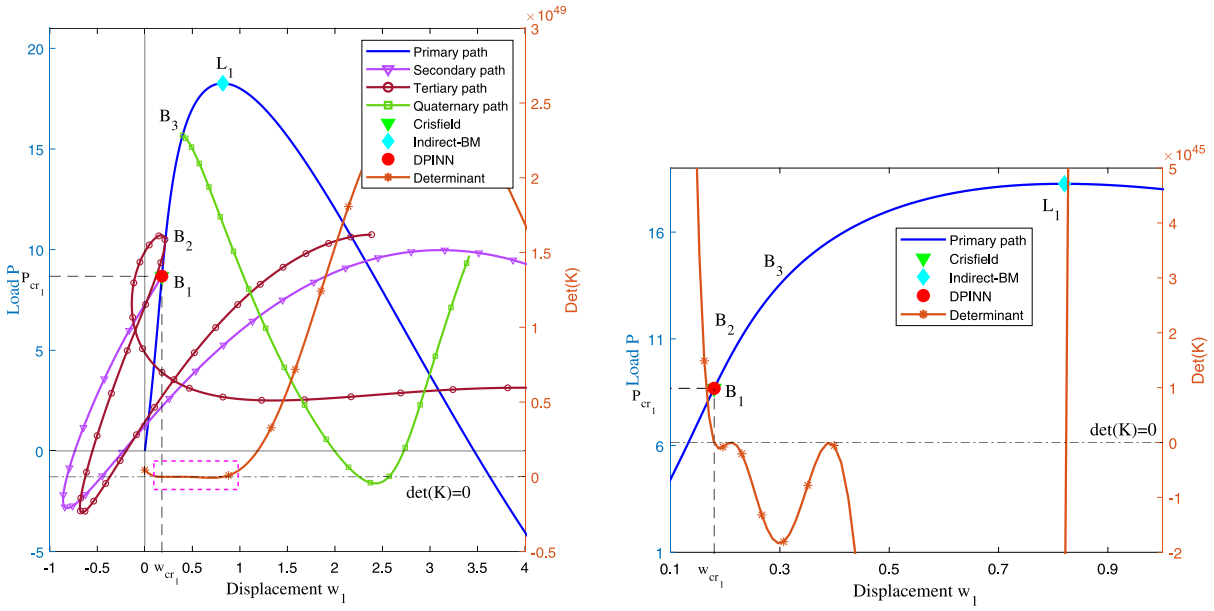

Fig. 20. Case 03: Load–deflection curve and critical points of the 24-bar dome truss.

Table 8
Loading conditions for dome truss with 24 bars.

Case	Nodes						
	1	2	3	4	5	6	7
Loading 1	P	0	0	0	0	0	0
Loading 2	P	P	P	P	P	P	P
Loading 3	P/2	P	P	P	P	P	P

Table 9
Comparison results obtained for 24-bar dome truss in searching the first critical point.

Results	Loading 1			
	Papadrakakis [60]	Oñate [20]	Indirect-BM	DPINN
w_{cr_1}	0.7610	0.7690	0.7690	0.7683
P_{cr_1}	3.1570	3.0960	3.1563	3.1564
Relative error (%)	–	1.8945	0.2462	0.2267
Time (s)	–	–	7.7660	59.5033
	Loading 2			
	Irles [61]	Oñate [20]	Indirect-BM	DPINN
w_{cr_1}	0.8820	0.8870	0.8766	0.8794
P_{cr_1}	7.8100	7.9700	7.7211	7.8111
Relative error (%)	–	2.0367	1.1326	0.0363
Time (s)	–	–	6.1286	51.4365
	Loading 3			
		Crisfeld [52]	Indirect-BM	DPINN
w_{cr_1}		0.1796	0.8207	0.1797
P_{cr_1}		8.6800	18.2628	8.6816
Relative error (%)		–	110.6244	0.0190
Time (s)		–	16.4905	56.2447

the first limit points obtained by two methods are found at the position on the equilibrium path where the determinant of the stiffness matrix vanishes. While the other results only approximate one of the two conditions. The second case is examined to evaluate the effect of large curvature change at the snap-through. From Fig. 19, it can easily be seen that the first singular point L_1 is a vertex of the primary path. It is obvious that the NN is very close to Irles [61] with a small relative error of 0.0363% and better than Oñate [20] (2.0367%) and indirect-BM (1.1326%). And the last case, the complex response involves the critical points which are closely located on the primary path, as shown in Fig. 20. As expected, the present work shows quite a good efficiency for the first critical point detection with a very small relative error of 0.019%. Meanwhile, the indirect-BM only found the limit point L_1 instead of the bifurcation point B_1. Although the DPINN requires a computational effort for training compared with the indirect-BM, the above results have proven the simplicity, efficiency, and robustness of the present approach for stability analysis.

5. Conclusions

In this article, a robust physics-informed NN framework is successfully developed for predicting structural instability. Accordingly, the network is built to parametrize the displacement field and load factor. In order to achieve the objective, a newly suggested loss function, which contains the residual load vector and property of the tangent stiffness matrix is established to guide the learning network. Therein, the parameters of the NN including weights and biases are design variables which are found by training to minimize the loss function corresponding to the position of the critical point. The simplicity, efficiency, and robustness of the proposed framework are demonstrated through several benchmark problems for the stability analysis of truss structure. Numerical results indicated that the first critical point attained by this work outperforms previously released works. Furthermore, the DPINN yields a simple and easily method to perform the stability analysis without utilizing any incremental-iterative techniques as well as structural analyses. In addition, one of the intriguing aspects of this paradigm is that its learning capability only relies upon the set of nodal coordinates which are known as the input data. Hence, the obtained results as well as the whole training data independent of the sampling techniques. Besides, the sensitivity analyses become easy and simple to implement by utilizing the automatic differentiation. In light of the above outstanding features, it is promising to offer a state-of-the-art approach without using numerical solvers to resolve complex issues in the structural stability analysis, such as thin-wall, plate, shell structures, etc.

CRediT authorship contribution statement

Hau T. Mai: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Tam T. Truong:** Data curation, Validation. **Joowon**

Kang: Data curation, Validation. **Dai D. Mai:** Writing review, Validation. **Jaehong Lee:** Conceptualization, Methodology, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgment

This research was supported by a grant (NRF- 2021R1A2B5B0300 2410) from NRF (National Research Foundation of Korea) funded by MEST (Ministry of Education and Science Technology) of Korean government.

References

[1] J. Shi, Computing critical points and secondary paths in nonlinear structural stability analysis by the finite element method, Comput. Struct. 58 (1) (1996) 203–220.
[2] P. Wriggers, J.C. Simo, A general procedure for the direct computation of turning and bifurcation points, Internat. J. Numer. Methods Engrg. 30 (1) (1990) 155–176.
[3] H. Sun, Y. Wang, W. Zhao, Comparison of theories for stability of truss structures. Part 1: Computation of critical load, Commun. Nonlinear Sci. Numer. Simul. 14 (4) (2009) 1700–1710.
[4] H. Sun, Y. Wang, Z. Wei, L. Chunliang, Comparison of theories for stability of truss structures. Part 2: Computation of critical solution of stability, Commun. Nonlinear Sci. Numer. Simul. 14 (4) (2009) 1711–1720.
[5] W. Wagner, P. Wriggers, Calculation of bifurcation points via fold curves, in: Nonlinear Computational Mechanics, Springer, Berlin, 1991, pp. 69–84.
[6] G. Skeie, C.A. Felippa, Detecting and traversing bifurcation points in nonlinear structural analysis, Int. J. Space Struct. 6 (2) (1991) 77–98.
[7] H.J. Weinitschke, On the calculation of limit and bifurcation points in stability problems of elastic shells, Int. J. Solids Struct. 21 (1) (1985) 79–95.
[8] E. Riks, The application of Newton's method to the problem of elastic stability, J. Appl. Mech. 39 (1972) 1060–1066.
[9] J. Simo, P. Wriggers, K. Schweizerhof, R. Taylor, Finite deformation post-buckling analysis involving inelasticity and contact constraints, Internat. J. Numer. Methods Engrg. 23 (5) (1986) 779–800.
[10] J. Shi, M. Crisfield, A semi-direct approach for the computation of singular points, Comput. Struct. 51 (1) (1994) 107–115.
[11] S.L. Chan, A non-linear numerical method for accurate determination of limit and bifurcation points, Internat. J. Numer. Methods Engrg. 36 (16) (1993) 2779–2790.
[12] M. Rezaiee-Pajand, H.R. Vejdani-Noghreiyan, A.R. Naghavi, Four new methods for finding structural critical points#, Mech. Based Des. Struct. Mach. 41 (4) (2013) 399–420.
[13] K. Ikeda, K. Murota, A. Yanagimoto, H. Noguchi, Improvement of the scaled corrector method for bifurcation analysis using symmetry-exploiting block-diagonalization, Comput. Methods Appl. Mech. Engrg. 196 (9–12) (2007) 1648–1661.
[14] H. Noguchi, F. Fujii, Eigenvector-free indicator, pinpointing and branch-switching for bifurcation, Commun. Numer. Methods. Eng. 19 (6) (2003) 445–457.
[15] J.P. Abbott, An efficient algorithm for the determination of certain bifurcation points, J. Comput. Appl. Math. 4 (1) (1978) 19–27.
[16] G. Moore, A. Spence, The calculation of turning points of nonlinear equations, SIAM J. Numer. Anal. 17 (4) (1980) 567–576.
[17] I. Planinc, M. Saje, A quadratically convergent algorithm for the computation of stability points: The application of the determinant of the tangent stiffness matrix, Comput. Methods Appl. Mech. Engrg. 169 (1–2) (1999) 89–105.
[18] A. Eriksson, Structural instability analyses based on generalised path-following, Comput. Methods Appl. Mech. Engrg. 156 (1–4) (1998) 45–74.
[19] P. Wriggers, W. Wagner, C. Miehe, A quadratically convergent procedure for the calculation of stability points in finite element analysis, Comput. Methods Appl. Mech. Engrg. 70 (3) (1988) 329–347.
[20] E. Oñate, et al., A critical displacement approach for predicting structural instability, Comput. Methods Appl. Mech. Engrg. 134 (1–2) (1996) 135–161.
[21] J.-M. Battini, C. Pacoste, A. Eriksson, Improved minimal augmentation procedure for the direct computation of critical points, Comput. Methods Appl. Mech. Engrg. 192 (16–18) (2003) 2169–2185.
[22] T.T. Truong, S. Lee, J. Lee, An artificial neural network-differential evolution approach for optimization of bidirectional functionally graded beams, Compos. Struct. 233 (2020) 111517.
[23] S. Wang, H. Wang, Y. Zhou, J. Liu, P. Dai, X. Du, M.A. Wahab, Automatic laser profile recognition and fast tracking for structured light measurement using deep learning and template matching, Measurement 169 (2021) 108362.
[24] W. Li, M.Z. Bazant, J. Zhu, A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches, Comput. Methods Appl. Mech. Engrg. 383 (2021) 113933.
[25] H.T. Mai, Q.X. Lieu, J. Kang, J. Lee, A robust unsupervised neural network framework for geometrically nonlinear analysis of inelastic truss structures, Appl. Math. Model. 107 (2022) 332–352.
[26] H.T. Mai, J. Kang, J. Lee, A machine learning-based surrogate model for optimization of truss structures with geometrically nonlinear behavior, Finite Elem. Anal. Des. 196 (2021) 103572.
[27] H.T. Mai, Q.X. Lieu, J. Kang, J. Lee, A novel deep unsupervised learning-based framework for optimization of truss structures, Eng. Comput. (2022) 1–24.
[28] H.T. Mai, S. Lee, D. Kim, J. Lee, J. Kang, J. Lee, Optimum design of nonlinear structures via deep neural network-based parameterization framework, Eur. J. Mech. A Solids (2022) 104869.
[29] A. Chandrasekhar, K. Suresh, TOuNN: Topology optimization using neural networks, Struct. Multidiscip. Optim. 63 (3) (2021) 1135–1149.
[30] H.T. Kollmann, D.W. Abueidda, S. Koric, E. Guleryuz, N.A. Sobh, Deep learning for topology optimization of 2D metamaterials, Mater. Des. 196 (2020) 109098.
[31] H. Wessels, C. Weißenfels, P. Wriggers, The neural particle method–An updated Lagrangian physics informed neural network for computational fluid dynamics, Comput. Methods Appl. Mech. Engrg. 368 (2020) 113127.
[32] T.T. Truong, D. Dinh-Cong, J. Lee, T. Nguyen-Thoi, An effective deep feedforward neural networks (DFNN) method for damage identification of truss structures using noisy incomplete modal data, J. Build. Eng. 30 (2020) 101244.
[33] Q.X. Lieu, K.T. Nguyen, K.D. Dang, S. Lee, J. Kang, J. Lee, An adaptive surrogate model to structural reliability analysis using deep neural network, Expert Syst. Appl. 189 (2022) 116104.
[34] S. Khatir, S. Tiachacht, C. Le Thanh, E. Ghandourah, S. Mirjalili, M.A. Wahab, An improved artificial neural network using arithmetic optimization algorithm for damage assessment in FGM composite plates, Compos. Struct. 273 (2021) 114287.
[35] L.V. Ho, T.T. Trinh, G. De Roeck, T. Bui-Tien, L. Nguyen-Ngoc, M.A. Wahab, An efficient stochastic-based coupled model for damage identification in plate structures, Eng. Fail. Anal. 131 (2022) 105866.
[36] L.V. Ho, D.H. Nguyen, M. Mousavi, G. De Roeck, T. Bui-Tien, A.H. Gandomi, M.A. Wahab, A hybrid computational intelligence approach for structural damage detection using marine predator algorithm and feedforward neural networks, Comput. Struct. 252 (2021) 106568.
[37] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[38] D.W. Abueidda, Q. Lu, S. Koric, Meshless physics-informed deep learning method for three-dimensional solid mechanics, Internat. J. Numer. Methods Engrg. 122 (23) (2021) 7182–7201.
[39] M.B. Gorji, M. Mozaffar, J.N. Heidenreich, J. Cao, D. Mohr, On the potential of recurrent neural networks for modeling path dependent plasticity, J. Mech. Phys. Solids 143 (2020) 103972.
[40] D.W. Abueidda, S. Koric, N.A. Sobh, H. Sehitoglu, Deep learning for plasticity and thermo-viscoplasticity, Int. J. Plast. 136 (2021) 102852.
[41] V.M. Nguyen-Thanh, X. Zhuang, T. Rabczuk, A deep energy method for finite deformation hyperelasticity, Eur. J. Mech. A Solids 80 (2020) 103874.
[42] J.N. Fuhg, N. Bouklas, The mixed deep energy method for resolving concentration features in finite strain hyperelasticity, J. Comput. Phys. 451 (2022) 110839.
[43] Q. Zhu, Z. Liu, J. Yan, Machine learning for metal additive manufacturing: Predicting temperature and melt pool fluid dynamics using physics-informed neural networks, Comput. Mech. 67 (2) (2021) 619–635.
[44] M. Yin, X. Zheng, J.D. Humphrey, G.E. Karniadakis, Non-invasive inference of thrombus material properties with physics-informed neural networks, Comput. Methods Appl. Mech. Engrg. 375 (2021) 113603.
[45] S.A. Niaki, E. Haghighat, T. Campbell, A. Poursartip, R. Vaziri, Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture, Comput. Methods Appl. Mech. Engrg. 384 (2021) 113959.
[46] N.N. Vlassis, W. Sun, Sobolev training of thermodynamic-informed neural networks for interpretable elasto-plasticity models with level set hardening, Comput. Methods Appl. Mech. Engrg. 377 (2021) 113695.
[47] R.G. Patel, I. Manickam, N.A. Trask, M.A. Wood, M. Lee, I. Tomas, E.C. Cyr, Thermodynamically consistent physics-informed neural networks for hyperbolic systems, J. Comput. Phys. 449 (2022) 110754.

[48] S. Jun, C.S. Hong, Buckling behavior of laminated composite cylindrical panels under axial compression, Comput. Struct. 29 (3) (1988) 479–490.
[49] Z. Waszczyszyn, Numerical problems of nonlinear stability analysis of elastic structures, Comput. Struct. 17 (1) (1983) 13–24.
[50] C. Felippa, Nonlinear finite element methods, ASEN 5017, course material, 1999.
[51] J.P. Abbott, et al., Numerical Continuation Methods for Nonlinear Equations and Bifurcation Problems, The Australian National University, 1977.
[52] Crisfield, Nonlinear Finite Element Analysis of Solids and Structures. Volume 2: Advanced Topics, Wiley, New York, NY (United States), 1996.
[53] M. Fujikake, A simple approach to bifurcation and limit point calculations, Internat. J. Numer. Methods Engrg. 21 (1) (1985) 183–191.
[54] F. Fujii, K. Ikeda, H. Noguchi, S. Okazawa, Modified stiffness iteration to pinpoint multiple bifurcation points, Comput. Methods Appl. Mech. Engrg. 190 (18–19) (2001) 2499–2522.
[55] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[56] D. Pecknold, J. Ghaboussi, T. Healey, Snap-through and bifurcation in a simple structure, J. Eng. Mech. 111 (7) (1985) 909–922.
[57] E. Oñate, H. Tschöpe, P. Wriggers, Combination of the critical displacement method with a damage model for structural instability analysis, Eng. Comput. (2001).
[58] Y.-y. Qian, J.-L. Batoz, A numerical stability study on truss structures, Revue Européenne Des Éléments Finis 1 (4) (1992) 461–479.
[59] H. Kwok, M. Kamat, L. Watson, Location of stable and unstable equilibrium configurations using a model trust region quasi-Newton method and tunnelling, Comput. Struct. 21 (5) (1985) 909–916.
[60] M. Papadrakakis, Post-buckling analysis of spatial structures by vector iteration methods, Comput. Struct. 14 (5–6) (1981) 393–402.
[61] R.I. Mas, Un Modelo Numerico Para El Analisis De Colapso En Entramados Metalicos (Ph.D. thesis), Universitat Politècnica de València, 1985.